UNIVERSITY OF LIVERPOOL | SCHOOL OF ENGINEERING

Emulation of complex dynamic models using Machine Learning Methods

Ryan David Jackson

School of Engineering
University of Liverpool
England

May 2021

A thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy

Abstract

Physical law based models are widely applied in the aerospace industry, providing models of dynamic systems such as rotorcraft flight simulators. To meet the criterion that such simulators need to run in real-time, simplifications to the underlying physics sometimes have to be applied, leading to errors in the model's predictions. To increase the accuracy of flight simulators, experienced test pilots and modellers are required to run through many iterations of models, an approach that requires significant time and cost. In the current thesis, a machine learning approach is used to develop an initial flight test model. Creating a more accurate initial model reduces the number of iterations required during model development and, as such, it is of interest to see whether machine learning can be used to realise sufficiently accurate initial models. The machine learning methods used are probabilistic and can capture the uncertainties associated with the model's predictions. Machine learning is facilitated using a Gaussian Process (GP) non-linear auto-regressive with exogenous inputs (NARX) model, which predicts the on-axis responses of pitch rate, roll rate, yaw rate and heave of a Bo105 rotorcraft. The auto-regressive with exogenous inputs (ARX) input structure and the parameter structure of the GP are investigated to determine which combination of input/parameter structure creates the most accurate predictions. Due to the nature of GPs they can be computationally expensive for large datasets and for a flight simulator model there would be a significant amount of data, especially if the GP model was trained on a full flight test campaign. A method that reduces the number of data points is therefore beneficial. In the current thesis, a variational sparse GP is used to reduce the computational cost of implementing the approach on large datasets by reducing the 'training' dataset of the GP. The results from the sparse GP are used to establish that the approach is scalable to larger datasets. Once the GP model is established, the idea of 'grey-box' approach is explored, combining the physical law based model ('white-box') and the GP model ('black-box'). The GP model, in the case of the grey-box approach, attempts to predict the error between the physical law based model and the true rotorcraft response. The hypothesis is that the grey-box approach may be able to function well in regions where there is little or no training data (an advantage over the purely machine-learnt data-based model). If the GP was applied to a full flight test campaign, it is likely that due to the sparsity of the training set the GP would have to make predictions in regions where there is little or no training data; in this case, the grey-box approach would have the advantage of relying on physical law based modelling assumptions, rather than only statistical patterns that had been identified in

the training data. The GP, the variational sparse GP and the grey-box model are compared to unseen flight test data to judge the efficacy of the models. The models are also compared to a current initial physical law based flight simulator model to investigate if the GP models can create more accurate predictions.

Acknowledgements

I would firstly like to thank Peter Green for his outstanding supervision during my PhD. His experience guided me along the journey and allowing me to research my ideas (we first tried to predict diesel engine emissions) has made the past five years a great experience. I will miss the bike chats after meetings. Hopefully, there will be more in the future. His love for machine learning has inspired me to seek a career in the field and to one day, maybe continue my research in machine learning. I would also like to acknowledge Mike Jump for allowing the use of the rotorcraft data and all of the advice on rotorcraft dynamics. His guidance has been invaluable. I have to thank my partner, Amy, who has had to deal with the ups and downs of the PhD, the stressful moments and being a constant support for me. Finally, I must thank my family; without their support, I would have never had the opportunity to go to university.

List of Publications

1. R. D. Jackson, M. Jump, and P. L. Green, "Towards Gaussian Process Models of Complex Rotorcraft Dynamics," in *Proceedings of the 74th Annual Forum of the Vertical Flight Society*, May 2018.

2. R. D. Jackson, M. Jump, and P. L. Green, "Predicting On-axis Rotorcraft Dynamic Responses Using Machine Learning Techniques," *Journal of the American Helicopter Society*, vol.65, no.3, pp.1–12, 2020. DOI: 10.4050/JAHS.65.032004.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Flight simulators form a vital part of any aircraft life cycle. They are used in design and development phases, testing and qualification activities as well as in training and research [1, 2]. Due to their availability and relatively low cost compared to the corresponding in-service aircraft, the use of simulators continues to increase. Flight training simulators, in particular, may help to address the increased demand for new pilots, who are needed to replace the current ageing population of pilots, by providing both initial and recurrent training opportunities. Moreover, the military is increasing the use of simulators for mission rehearsal in land, sea and air contexts. The study by Bell and Waag [2] documents the effectiveness of training combat skills in flight simulators.

The fidelity of any flight simulation experience reflects the accuracy of the flight model and the realism of the other integrated components of the simulator. The current thesis focusses on the first of these elements, the flight dynamics model. Techniques to design and develop such models are well-known and documented [3, 4]. Of course, to be of any utility, the entire simulation system must run in real-time. This requirement can necessitate simplifications to the model's underlying physics having to be made, particularly when complex aircraft, such as rotorcraft, is the subject of interest. These simplifications mean that the flight model cannot necessarily capture all of the complex dynamics that would be present during the equivalent real

scenario, potentially leading to significant differences between the model and the real aircraft. These differences can, in the worst case, have a negative impact on training for the crew using the simulator.

The quality of the flight dynamics model, therefore, clearly influences the fidelity of the simulation device. The 'engineering fidelity' of such a device is typically measured against a series of quantitative requirements contained within simulator qualification documents such as [5, 6]. It is recognised that examining the response of the simulator in this way only partially serves to characterise its utility. While efforts are underway to develop methods that can better meet this need [7, 8, 9, 10], the thesis seeks to explore techniques whereby the accuracy of the flight dynamics element of the simulation device can be improved, even when the modelled physics can no longer accurately represent reality.

Current approaches to flight simulator model development require a large number of iterations. The first iteration is to create an initial physical law based model of the rotorcraft; the model is then tested by a test pilot who suggests corrections. The model is altered according to the suggestions, and iterations through the testing and corrections process are performed until the solution converges. These iterations can take a substantial amount of time and require an experienced test pilot as well as a flight simulator modelling team. The simulator is assessed against rating scales, such as the Cooper-Harper handling qualities [11] or the rating scales developed by Perfect et al. [7], which are subjective. The main benefit of the current approach to flight simulator model development is that an experienced pilot guides the development of the rotorcraft simulation model; the model would, therefore 'feel' like a real rotorcraft. The work presented in the thesis attempts to remove a large number of these iterations and to create a more accurate initial model. Decreasing the number of iterations would reduce time and cost, as the experienced pilot and flight simulator modelling team would only be required to 'fine-tune' the model.

1.2 Objectives

The work in the thesis focusses on creating a data-based 'black-box' and 'grey-box' model for flight simulation purposes. The data-based method could potentially reduce the time complex-

ity of the development stages of the current flight simulator model life cycle. A data-based model has the potential to be trained relevantly quickly and could even be higher fidelity than the initial physical law based model. While methods are being developed to enhance physical law based models [12], the use of the data-based model could reduce the number of pilot assessment iterations required, due to the higher fidelity potential. The data generated from flight test campaigns is not necessarily fully exploited, and the data-based model would use this data to generate a representative model of the true aircraft. It is envisaged that the proposed 'black-box' and 'grey-box' models would be best utilised when the flight simulator role is training and mission rehearsal, due to the range of the flight envelope being restricted.

Machine learning - the development of models, based on patterns and/or correlations found in available data - has the potential to improve current flight simulators. The field of machine learning has expanded rapidly in recent years, as a result of the increased abundance of data in many disciplines. While allowing the exploitation of these data, the models derived using such approaches are generally only valid in, or close to, the regions where data is available.

In the current thesis, a Gaussian Process (GP) is used to facilitate machine learning. GPs are probabilistic non-parametric regression methods that have been widely studied in the machine learning community(see Williams and Rasmussen [13], Rasmussen [14] and Williams [15]). In the current thesis, GP models are developed to predict pitch rate, roll rate, yaw rate and heave for a Bo105 rotorcraft. These predictions are compared with both unseen data from the true rotorcraft response (i.e. data not used in training) and to an initial physical law based model predictions, from a rotorcraft flight dynamics model implemented using Advanced Rotorcraft Technology's (ART) FLIGHTLAB software [16]. It is investigated if it is feasible to use auto-regressive with exogenous inputs (so-called 'ARX') as the input structure for the GP model of the rotorcraft. To test if a simpler model can capture the rotorcraft dynamics, linear regression is first used to produce models. The linear regression models are unable to capture the dynamics of the rotorcraft with sufficient accuracy. It was found that the GP with an ARX input structure created the most accurate predictions. The GP models using the ARX input structure predicting the on-axis responses are explored for the pitch rate, roll rate, yaw rate and heave. It is unclear if off-axis pilot inputs affect the GP predictions; therefore, different

ARX input structures are utilised (which include all of the current pilot inputs).

One of the known disadvantages of GPs is that they can be computationally expensive if a large training dataset is used. It is, therefore, beneficial to be able to reduce the number of data points used when training a GP algorithm. The aim is to investigate whether a method to reduce the size of the GP training data would be scalable to larger datasets. In this thesis, the variational sparse GP approach [17, 18] is used to automatically create a 'sparse GP' (a GP trained on a subset of the available data) that closely approximates a GP that has been trained on all of the available data. The sparse GP model is compared to the true response of the rotorcraft to investigate whether the sparse GP can produce an accurate model despite being trained on only a fraction of the available data. The physical law based model is compared to the sparse GP to see if, even with a small training dataset, the sparse GP can outperform the physical-law based model.

Once a machine learnt flight model has been established, the aim is to investigate a 'grey-box' model. A grey-box model attempts to combine the physical law based equations ('white-box' model), and a machine learnt flight model ('black-box' model). The black-box model in the current thesis is a GP predicting the error between the true rotorcraft response and the physical law based model. An advantage of the grey-box approach compared to the purely machine learnt flight model is that the grey-box model may be able to perform well in regions where there is little or no training data.

In summary, the thesis investigates:

- Firstly, if a linear regression model can capture the dynamics of the on-axis pitch rate response of the Bo105 rotorcraft.
- Whether using a machine learning approach, it is possible to predict the pitch rate, roll rate, yaw rate and heave of a rotorcraft with a level of fidelity that is comparable to what can be achieved using a physical law based model (FLIGHTLAB, in this case).
- The most appropriate input/parameter structure for the proposed machine learning approach, given the longitudinal stick position, lateral stick position, pedal position and

collective lever data available (the inputs are discussed in more detail in Chapter 3).

- If the uncertainty in the optimisation of the GP hyperparameters will affect the accuracy of the GP models.
- The *scalability* of the proposed variational sparse GP approach, envisaging future applications where much larger sets of training data will likely be utilised. It is investigated whether the use of the variational sparse GPs [17, 18] can significantly reduce the cost of applying the proposed method to much larger datasets.
- If the grey-box approach can be used to accurately predict the error between the physical law based model and the true rotorcraft response.

A summary of the main outcomes is as follows:

- The thesis displays evidence that GPs have the potential to improve the accuracy of the initial flight test models compared to existing physical law based models.
- The use of ARX input structure is required to accurately capture the dynamics of pitch rate, roll rate, yaw rate and heave.
- All of the current pilot inputs are required to capture the dynamics of the pitch rate, roll rate, yaw rate and heave.
- For the pitch, roll, yaw rate GP models, the ARX input structure required two lagged responses terms to capture the dynamics of the response.
- The uncertainty of the optimisation for the GP parameters does not have a large effect on the GP predictions.
- Sparse GPs have the potential to scale the approach to much larger datasets; which will be required if the approach was to be implemented across a flight test campaign.
- The grey-box model's predictions were not as accurate as the purely data-based GP models; however, the grey-box model did produce more accurate predictions than the initial physical law based model.

1.3 Outline of Chapters

The thesis layout is in the following format:

- Chapter 2 introduces the theory of GP regression, the optimisation techniques for the parameters and the use of different ARX GP input structures.
- Chapter 3 presents the Bo105 rotorcraft, the physical law based FLIGHTLAB model and explains the datasets used for the GP predictions in detail.
- Chapter 4 displays a feasibility study regarding the Bo105 rotorcraft and provides evidence that the GP regression models using an ARX input structure provide more accurate predictions than the linear regression models and in fact can capture the on-axis dynamics of the Bo105 rotorcraft for the pitch rate.
- Chapter 5 builds on the results presented in Chapter 4 and investigates different ARX input structures for each of the rotorcraft responses (pitch rate, roll rate, yaw rate and heave). The GP results from the ARX structures, for each rotorcraft response, are compared to examine which ARX input structure provides the most accurate predictions compared to the true rotorcraft response.
- Chapter 6 builds on the results found in Chapter 5 and investigates if assigning a parameter to each input of the ARX input structure creates more accurate GP predictions. Combining the results from Chapters 5 and 6, it is then clear which GP input and parameter structure produce the most accurate predictions, compared to the true rotorcraft response, for each rotorcraft axis.
- Chapter 7 provides the theory of the variational sparse GP, which is a method that attempts to reduce the number of training points required to make accurate predictions.
- Chapter 8 uses the variational sparse GP approach and applies it to the Bo105 rotorcraft for the on-axis response of the pitch rate, roll rate, yaw rate and heave. The GP results from Chapters 5 and 6 display which input/parameter structure produces the most accurate GP predictions, compared to the true rotorcraft response, for each of the rotorcraft

axes. Using the knowledge gathered from Chapters 5 and 6, the input/parameter structure that produced the more accurate predictions is used for the sparse GPs. The sparse GP predictions are compared to the true rotorcraft response as well as the predictions made by the GP (Chapters 5 and 6) and the physical law based model.

- Chapter 9 uses the grey-box approach, to investigate if the GP can predict the error between the physical law based model and the true rotorcraft response for the on-axis response of pitch rate, roll rate, yaw rate and heave. The grey-box approach is compared to the GP (Chapters 5 and 6) and the variational sparse GP predictions to examine if the grey-box or black-box (data-based GPs) approach creates the most accurate predictions compared to the true rotorcraft response.

- Chapter 10 provides a summary of the thesis achievements and suggestions for future work.

Chapter 2

Background Theory

Gaussian Processes (GPs) have been widely used in recent years and applied to many different applications, for example in engineering applications, where the GP was used to model an internal combustion engine [19, 20, 21, 22] and model an autonomous blimp [23]. In Berger et al. [19] the GP is used to create surrogate models for engine calibration (emissions) and to reduce the time required on the testbed. Berger and Rauscher [22] used the GP for diesel engine calibration due to the outliers within the data, the GP model displayed is a robust GP, in which the outliers do not affect the GP model. The studies by Castric et al. [20] and Gutjahr et al. [21] used a GP model to identify the emissions of a diesel engine, where both studies created a fast emulator. Castric et al. compared two models, geostatistics and a GP, and concluded that the GP produce more accurate models of the emissions.

The use of GPs in the statistical community started in 1978, when O'Hagan [24] used the idea of defining a prior distribution over a function, and applied this to a curve-fitting example. GP for regression was introduced into the machine learning field in 1996 by Williams and Rasmussen [13], other notable early work by Rasmussen [14], Neal [25] and Williams [15] all introducing the idea of GP regression into the machine learning field. Since the introduction of GPs to the machine learning community, there has been much-published work on the theory of GP regression. The main contributor to the GP theory in the current thesis is the book "Pattern Recognition and Machine learning" by Bishop [26]; the book covers all the concepts from a probability distribution to kernel methods and sampling methods. Chapter one of the thesis

by Boyle [27] introduces the GP regression theory in detail. The GP theory for regression is also given in work by Seeger [28], Rasmussen [29, 30] and Ebden [31].

In the current thesis, GPs are used to perform regression (they also can perform another task such as classification [26]). An advantageous property of GPs is that they can be used to quantify the uncertainties in one's predictions which, in this case, produces a worst-case scenario given the model uncertainty. It is envisaged that the uncertainties in the predictions could be used in the development stage of the GP model, to highlight areas in which more training data is required to make accurate predictions. Using the uncertainties to assist the development of the GP model is not shown in the current thesis but could provide scope for future work and is discussed in more detail in Section 10.2. Another beneficial property of GPs is that once trained; they can produce very fast emulators of complex models. Producing fast emulators is advantageous when dealing with non-linear behaviours, such as rotorcraft dynamics.

2.1 Background Theory of Regression Models

In the current section, an introduction to linear regression is given and followed by a brief background to the GP theory and the optimisation processes used in the thesis. The section commences with the theory of linear regression, which is used to create clarity on the GP regression theory. The GP regression theory is followed by the theory of the optimisation processes used in the thesis. After the optimisation processes are defined, the input structure, the choice of kernel and the number of parameters used for the GP are discussed.

2.1.1 Linear Regression

Suppose one seeks to learn a function that maps inputs ($\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$) to observed outputs ($\boldsymbol{y} = [y_1, \ldots, y_N]^T$), where in the context of the current thesis, $\boldsymbol{x}$ may represent a pilot input and y the response of the rotorcraft (e.g. pitch rate). Consider a regression model that is defined as a linear combination of basis functions [26, 29]:

$$f_i = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) \tag{2.1}$$

where $\boldsymbol{w}$ is a vector of parameters to be identified, and $\boldsymbol{\phi}(\cdot)$ is a basis function. Note that f_i is equivalent to $f(x_i)$.

For example, a second order polynomial may result in the parameter vector $\boldsymbol{w}$ and basis function $\boldsymbol{\phi}(x)$ being expressed by:

$$\boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad \boldsymbol{\phi}(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}. \tag{2.2}$$

The aim is to find optimum parameters, $\boldsymbol{w}^*$, that minimise the squared error between observations $\boldsymbol{y}$ and predictions $\boldsymbol{f}$:

$$J = \sum_{i=1}^{N} (y_i - f_i)^2. \tag{2.3}$$

As the model in equation (2.1) is linear in $\boldsymbol{w}$, one can realise a closed form expression for $\boldsymbol{w}^*$, firstly defining:

$$\boldsymbol{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} \tag{2.4}$$

which may be expressed as

$$\begin{aligned} J &= (\boldsymbol{y} - \boldsymbol{f})^T (\boldsymbol{y} - \boldsymbol{f}) \\ &= \boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{y}^T \boldsymbol{f} + \boldsymbol{f}^T \boldsymbol{f} \\ &= \sum_i y_i^2 + 2\sum_i y_i f_i + \sum_i f_i^2 \\ &= \sum_i y_i^2 + 2\sum_i y_i \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + \sum_i \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{w} \end{aligned} \tag{2.5}$$

such that

$$\frac{\partial J}{\partial \boldsymbol{w}} = -2\sum_i y_i \boldsymbol{\phi}(\boldsymbol{x}_i) + 2\sum_i \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{w}. \tag{2.6}$$

Setting $\partial J/\partial \boldsymbol{w}$ equal to zero, it can be shown that:

$$\boldsymbol{w}^* = \left[\sum_i \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^T\right]^{-1} \left[\sum_i y_i \boldsymbol{\phi}(\boldsymbol{x}_i)\right]. \tag{2.7}$$

Regression models can have different basis functions, $\boldsymbol{\phi}(\boldsymbol{x})$ which affect the performance of the model. The input structures investigated in the thesis include moving average (MA), auto-regressive (AR), auto-regressive with exogenous inputs (ARX), auto-regressive moving average (ARMA) and auto-regressive moving average with exogenous inputs (ARMAX). The input structures are defined as follows:

- Moving Average (MA) - a linear combination of past inputs [32].
- Auto-regressive (AR) - a linear combination of past observations.
- Auto-regressive with exogenous inputs (ARX) - a linear combination of past observations, including the current inputs, which are described as exogenous inputs.
- Auto-regressive moving average (ARMA) - combines the auto-regressive and moving average input structures. The input structure, therefore, contains a linear combination of past inputs and observations.
- Auto-regressive moving average with exogenous inputs (ARMAX) - combines auto-regressive with exogenous inputs and moving average input structures. The input structure contains a linear combination of past inputs and outputs with the current inputs.

Each one of the input structures is investigated in Chapter 4 to display if a particular input structure creates more accurate predictions than the others.

2.1.2 From Linear Regression to Gaussian Process Regression

To aid the understanding of a GP approach, consider a regression model that is defined in equation (2.1):

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{2.8}$$

where $\boldsymbol{\phi}$ is a vector of 'basis functions' ($\boldsymbol{\phi} = [\phi_1, \ldots, \phi_D]$) and D is the number of 'basis functions'. The 'basis function' is limited to a parametric family, for example, a polynomial. If the target response does not belong to the same family as the 'basis function', the prediction may struggle to match the target. Examples of polynomial 'basis functions' are shown in Figure 2.1. A 'basis function' is a user-defined function, and there are many possible choices, an example is given in equation (2.2).

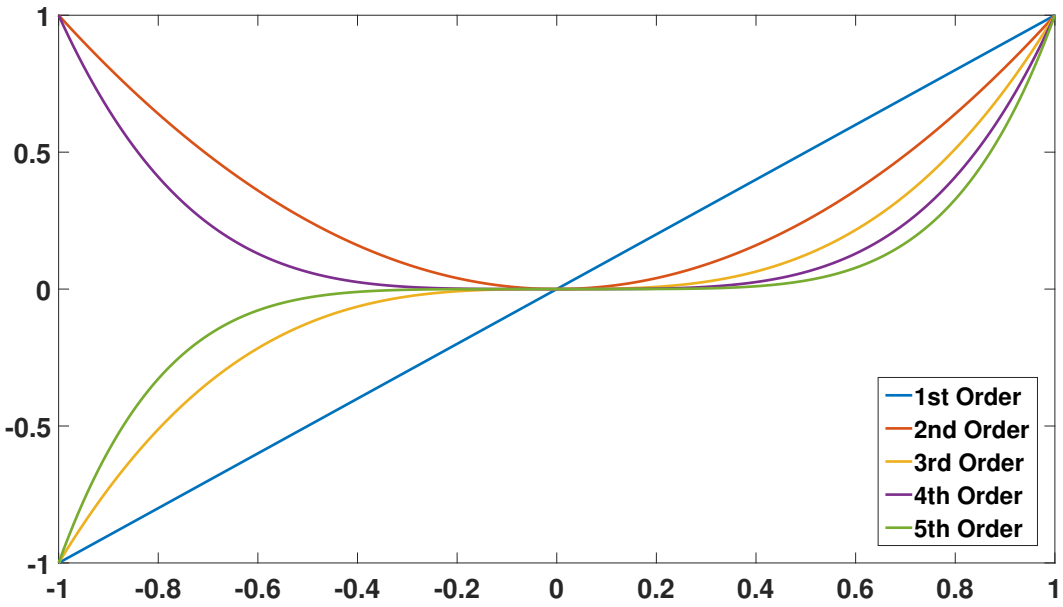


Figure 2.1: Examples of polynomial basis functions

Instead of the approach shown Section 2.1.2, one could adopt a Bayesian approach to the identification of $\boldsymbol{w}$; one could, for example, choose the prior distribution over $\boldsymbol{w}$ to be

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \mathbf{0}, \gamma^{-1}\boldsymbol{I}) \tag{2.9}$$

where γ is the precision of the distribution, and $\boldsymbol{I}$ is an identity matrix. The function values at the points where training data are available are given by vector $\boldsymbol{f}$

$$\boldsymbol{f} = \begin{pmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_N) \end{pmatrix} \tag{2.10}$$

where N is the number of training points, for any value of $\boldsymbol{w}$ defines a function of $\boldsymbol{x}$, given in equation (2.8). At this point, one would usually attempt to infer $\boldsymbol{w}$ from some observations, where in this case, the observations are the input-output pairs $\{(x_n, y_n)\}_{n=1}^{N}$.

The joint distribution of function values shown in equation (2.10) are of interest. Using equation (2.8), the vector $\boldsymbol{f}$ is given by:

$$\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{w} \tag{2.11}$$

where $\boldsymbol{\Phi}$ is referred to as the 'design matrix'. The 'design matrix' compromises of an $N \times D$ matrix whose elements are 'basis functions'.

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(x_1), & \phi_2(x_1) & \dots & \phi_D(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_N), & \phi_2(x_N), & \dots & \phi_D(x_N)) \end{bmatrix} \tag{2.12}$$

.

Given the prior in equation (2.9), one can then show that the prior distribution over $\boldsymbol{f}$ is Gaussian, with a mean

$$\mathbb{E}[\boldsymbol{f}] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}] = \boldsymbol{0} \tag{2.13}$$

and covariance matrix

$$\text{cov}[\boldsymbol{f}] = \mathbb{E}[\boldsymbol{f}\boldsymbol{f}^T] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{\Phi}^T = \frac{1}{\gamma}\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{K} \tag{2.14}$$

where $\boldsymbol{K}$ is a matrix with elements

$$K_{nm} = \frac{1}{\gamma}\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m); \qquad n, m = 1, \ldots, N. \tag{2.15}$$

One can then assume a noise model where the observations of the system's outputs are equal to the true response with the addition of some Gaussian noise. In other words, it is assumed that

$$y_n = f_n + \epsilon_n \tag{2.16}$$

where y_n represents the nth observation of the system's response, $f_n \equiv f(x_n, w_n)$ and $\epsilon_n \sim \mathcal{N}(\epsilon_n \mid 0, \beta^{-1})$, where β is the precision of the noise that corrupted the observation. If one defines the likelihood using the noise model

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}) \tag{2.17}$$

using the knowledge of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{\Phi}\boldsymbol{w}$, the likelihood can be written as

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\Phi}\boldsymbol{w}, \beta^{-1}) \tag{2.18}$$

therefore using the properties of Gaussian functions and knowledge of the prior in equation (2.9) and the likelihood in equation (2.18), the posterior is defined as

$$p(\boldsymbol{w} \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.19}$$

where

$$\boldsymbol{\Sigma} = (\gamma\boldsymbol{I} + \boldsymbol{\Phi}^T\beta\boldsymbol{\Phi})^{-1}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\Phi}^T\beta\boldsymbol{y}$$

one could now try to evaluate the posterior over $\boldsymbol{w}$. As an alternative, one can start to define a GP, instead of defining basis functions, as in equation (2.8), one can define a prior over $\boldsymbol{f}$ directly, such that

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f} \mid \mathbf{0}, \boldsymbol{K}_{NN}) \tag{2.20}$$

where the covariance matrix, $\boldsymbol{K}_{NN}$ is given by

$$\boldsymbol{K}_{NN} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \dots & k(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \vdots & \vdots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & k(\boldsymbol{x}_N, \boldsymbol{x}_2) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

and where (k) is a user-defined 'kernel function'. The kernel function is chosen so that $\boldsymbol{K}$ is a valid covariance matrix (i.e. symmetric and positive-semidefinite). An example of such a kernel function is the squared exponential [26]

$$\boldsymbol{K}_{NM} = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \exp\left(-\frac{\alpha}{2}(\boldsymbol{x}_n - \boldsymbol{x}_m)^T(\boldsymbol{x}_n - \boldsymbol{x}_m)\right) \tag{2.21}$$

where the 'hyperparameter' (α) have an effect on the smoothness of the kernel (shown in Figure 2.2) and induces correlations that depend on the closeness of $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$. When using the 'basis function', one is limited to a single parametric family, while with a GP one is no longer limited to a single parametric family (which is why GPs are sometimes referred to as 'non-parametric').

Choosing different hyperparameters can affect how accurate the GP model is and, as such, the hyperparameters are typically inferred from the observation data as part of the training process. The optimum hyperparameters are most commonly identified by maximising the likelihood of

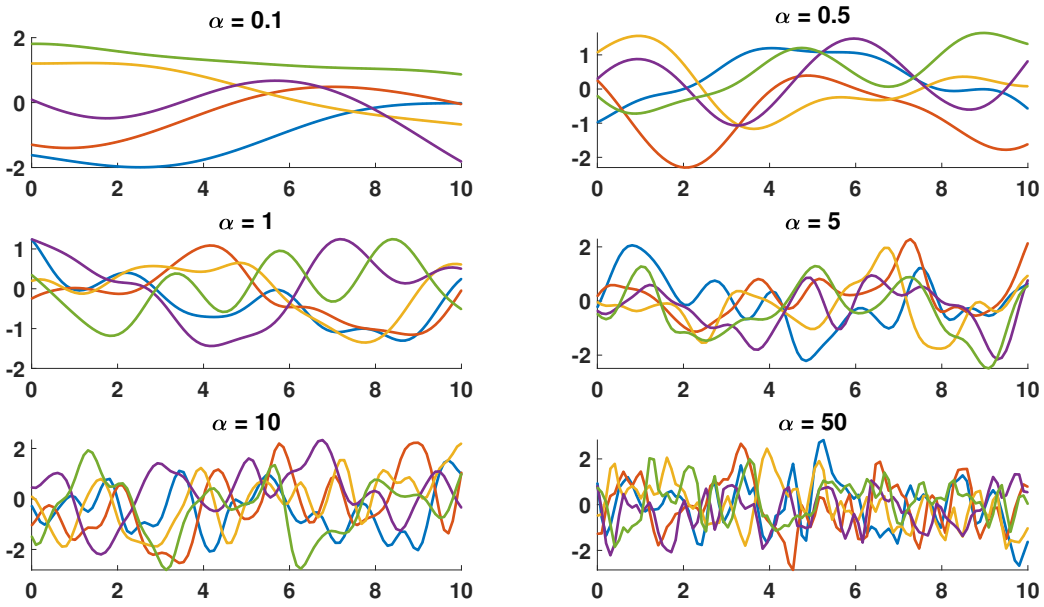


Figure 2.2: Examples of how the hyperparameter α affects the kernel function

witnessing the observed data (optimisation techniques are discussed further in Section 2.1.3, given by

$$p(\boldsymbol{y} \mid \boldsymbol{f}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \beta^{-1}\boldsymbol{I}). \tag{2.22}$$

Marginalising the likelihood over $\boldsymbol{f}$ gives

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{f}) p(\boldsymbol{f}) d\boldsymbol{f} \tag{2.23}$$

which, given equations (2.20) and (2.22), allows one to write

$$p(\boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{0}, \boldsymbol{C}) \tag{2.24}$$

where the elements of $\boldsymbol{C}$ are given by

$$C_{nm} \equiv C(\boldsymbol{x}_n, \boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^{-1} I_{nm} \tag{2.25}$$

where I_{nm} is the (n, m)th element of the identity matrix $\boldsymbol{I}$.

2.1.3 Optimisation of Hyperparameters for the Creation of Gaussian Processes

The GP training process requires the optimisation of the hyperparameters, the thesis by Boyle [27] and the book by Bishop [26] provide many optimisation techniques such as Markov Chain Monte Carlo (MCMC). Examples of optimisation techniques include gradient-based ascent methods [26] or population-based algorithms (SADE [33]). Both of these can be used to realise maximum likelihood estimates of the hyperparameters by maximising the log-likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = -\frac{1}{2}(N \log 2\pi + \log |\boldsymbol{C}| + \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y}). \tag{2.26}$$

where $\boldsymbol{\theta} = [\alpha, \beta]^T$. Alternatively, one can generate samples from the posterior parameter distribution $p(\boldsymbol{\theta} \mid \boldsymbol{y})$, which is given by Bayes' rule

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \tag{2.27}$$

where $p(\boldsymbol{\theta})$ represents the prior and $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ the likelihood. There are many different methods in which using samples from the posterior parameter distribution could be used, the techniques used to locate the optimum hyperparameters in the current thesis are discussed in the subsequent subsections.

Markov Chain Monte Carlo (MCMC)

Markov chain Monte Carlo (MCMC) can be used to generate samples from the posterior hyperparameter distribution, thus quantifying uncertainties in the hyperparameter selection. It can also provide a more global search, unlike the local search provided by gradient-ascent methods.

The best known MCMC method is the Metropolis algorithm (MA) [34]. In the following, the target distribution, $\pi(\boldsymbol{\theta})$ (i.e. the distribution from which one wishes to generate samples), is equal to $p(\boldsymbol{\theta} \mid \boldsymbol{y})$.

The first step in each iteration of the MA algorithm is to propose a new 'state' $\boldsymbol{\theta}'$, where the current state of the Markov chain is $\boldsymbol{\theta}^{(r)}$ [35]. The proposal is taken from a probability density function, $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(r)})$, which is conditional on the current state. The proposal is accepted as the new state of the Markov chain with probability

$$\min\left\{1, \frac{\pi(\boldsymbol{\theta}'))}{\pi(\boldsymbol{\theta}^{(r)})}\right\} = \min\left\{1, \frac{\pi^*(\boldsymbol{\theta}'))}{\pi^*(\boldsymbol{\theta}^{(r)})}\right\} \tag{2.28}$$

where $\pi^*(\boldsymbol{\theta})$ is the un-normalised target distribution. The normalised target distribution would have an unknown normalising constant. Therefore using the un-normalised target distribution removes the requirement to find the value of the unknown normalising constant. If accepted, the new state of the Markov chain is $\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}'$, otherwise $\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}^{(r)}$. This process is run for a user-defined number of samples. The Markov chain will then reach a stationary distribution producing posterior samples of the hyperparameters.

A simple example of a GP fitted to a sine wave is shown in Figure 2.3, where MCMC is used to find the optimal hyperparameters. The data used to create the sine wave is given by

$$\boldsymbol{y} = \sin(\boldsymbol{x}) + \epsilon \tag{2.29}$$

wherein this example ϵ is an error sampled from a normal distribution ($\mathcal{N}(0, 0.1)$) and $\boldsymbol{x}$ is a vector of 100 points linearly spaced between 0 and 2π. The sine data and the GP predictions are shown in Figure 2.3.

Figures 2.4 and 2.5 show the results of an MCMC accepted samples from the posterior parameter distribution

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{2.30}$$

where $\boldsymbol{\theta} = [\alpha, \beta]^T$.

The accepted samples in Figures 2.4 and 2.5 reach a stationary distribution (right-hand side of Figures 2.4 and 2.5), which is defined as when the distribution does not change for some

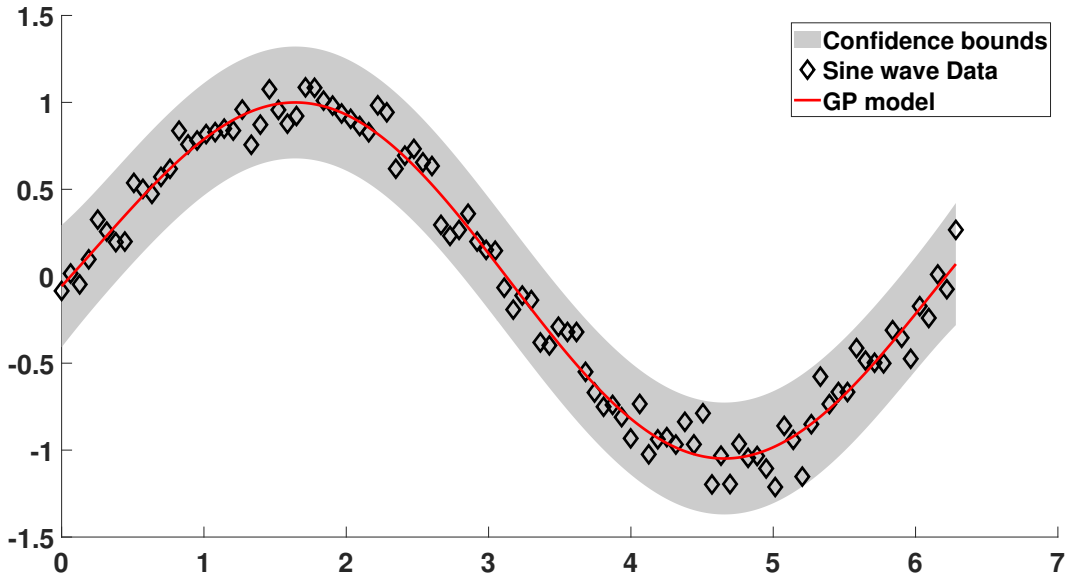


Figure 2.3: GP predictions for a sine wave data described in equation (2.29) and hyperparameters optimised by the MCMC results shown in Figures 2.4 and 2.5.

time. The accepted samples before reaching the stationary distribution is called 'burn-in'. The 'burn-in' is not used to calculate the hyperparameters. The stationary distribution is used to select the hyperparameters for the kernel function used to create the GP model shown in Figure 2.3. The hyperparameters are calculated by using the mean of the accepted samples once the MCMC has reached a stationary distribution, it can be hard to judge when the stationary distribution has occurred.

The selection of the proposal width is important if the MCMC is able to reach a stationary distribution; the proposal width is defined as the range in which the MCMC approach takes samples. In the case of the GP, the samples are hyperparameters. If the proposal is too broad, the MCMC approach will take a long time to reach a stationary distribution. A too narrow proposal width, the stationary distribution will more than likely be missed, and the optimal hyperparameters would not be located. Methods to reduce the effect of the initial choice of the proposal width are shown in Sections 2.1.3 and 2.1.5.

Simulated Annealing

Simulated annealing is a form of MCMC which uses the Metropolis algorithm (MA). The difference is that the method slowly increases the influence of the likelihood $p(\boldsymbol{y} \mid \boldsymbol{\theta})$. Effectively

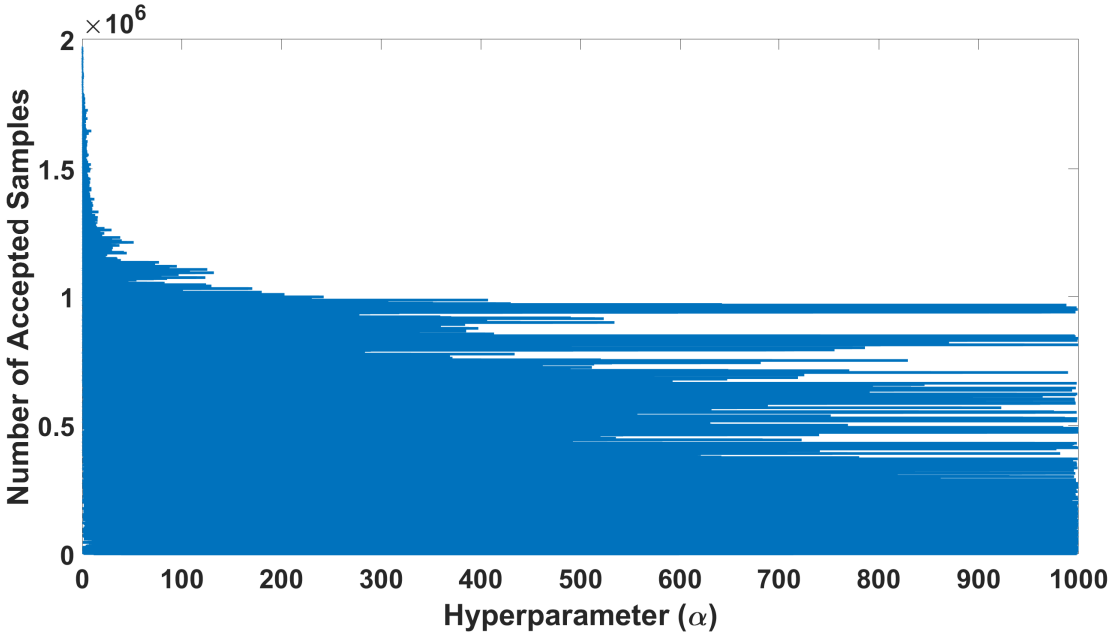


Figure 2.4: MCMC results for hyperparameter α for the GP predictions shown in Figure 2.3.

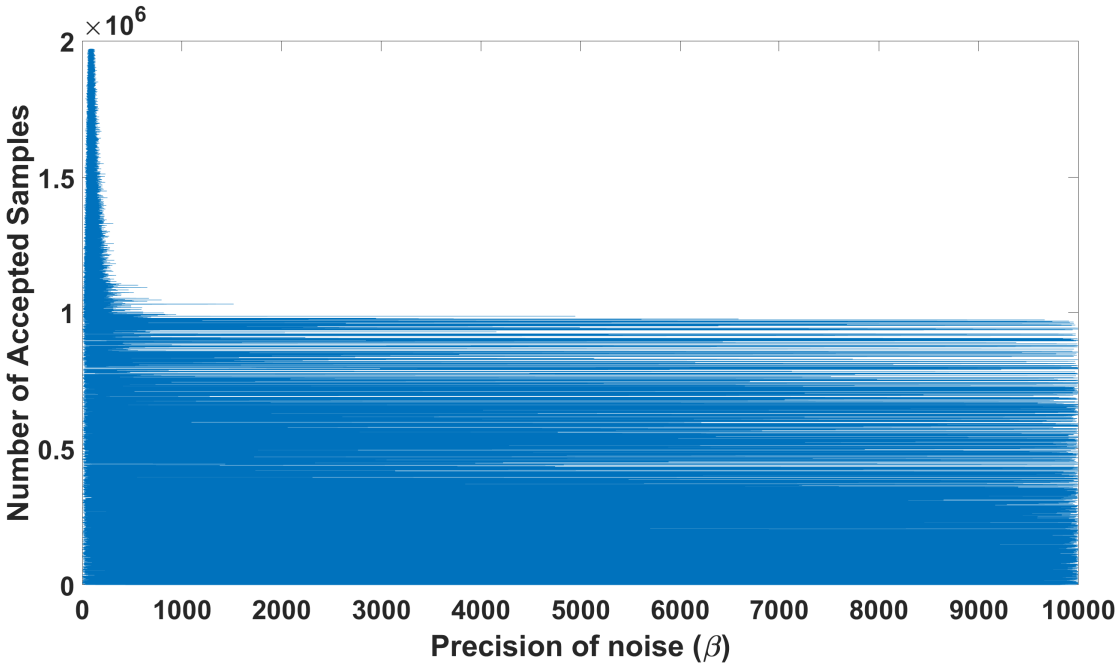


Figure 2.5: MCMC results for the precision of noise β for the GP predictions shown in Figure 2.3.

reducing the size of the proposal width during the optimisation, this reduces the problems discussed in Section 2.1.3 regarding the careful selection of the proposal width and the judgement of reaching the stationary distribution [36]. The target distribution has the form

$$\pi_{\zeta_j} \propto p(\boldsymbol{y} \mid \boldsymbol{\theta})^{\zeta_j} p(\boldsymbol{\theta}) \qquad\qquad j = 1, 2, \ldots, N_\zeta \tag{2.31}$$

where

$$0 = \zeta_1 < \zeta_2 < \cdots < \zeta_{N_\zeta} = 1. \tag{2.32}$$

Increasing the parameter ζ is inducing a transition from the prior to posterior parameter distribution [37]. The technique has been used as a fundamental part of transitional MCMC [38] and asymptotically independent Markov sampling [39] algorithms.

A small initial value of ζ is chosen close to zero but not equal to zero. By choosing a small initial ζ it is ensuring that the first target distribution is similar to the prior. For each ζ value, a full run of the MA is conducted using a user-defined number of samples R. Using the previously accepted samples from the MA, ζ is updated for the next run using the following equation

$$\zeta(j+1) = \zeta(j) + \frac{1}{\zeta(j) \times var(p(\boldsymbol{y} \mid \theta)))} \tag{2.33}$$

where the variance of the likelihood uses all the accepted samples from the previous run of the MA and j defines the previous run of the MA. The optimisation stops when ζ reaches one, in which the whole likelihood has now been introduced. The ζ update in equation (2.33) uses a constant change in Shannon entropy [37].

The advantage of using this method over the standard MA is that it is more user-friendly and more efficient, as the proposal width is being updated after each run of the MA. The standard MCMC is very dependent on the user-defined steps in which the steps could be too small (computationally expensive) or too large (could miss the optimal value).

Figures 2.7 and 2.8 show the results of a simulated annealing optimisation reaching a stationary

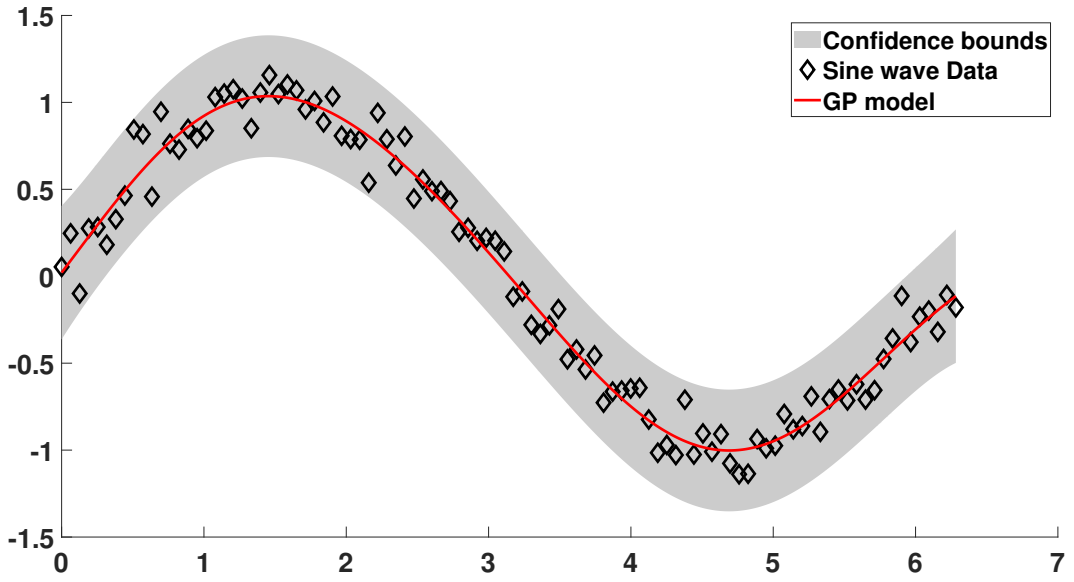


Figure 2.6: GP predictions for a sine wave data described in equation (2.29) and hyperparameters optimised by the SA results shown in Figures 2.7 and 2.8.

distribution. As the influence of the likelihood is increased the optimisation 'focusses' on the optimal solution. The stationary distributed is denoted as the accepted samples, and the burnt samples before the stationary distribution are shown in Figures 2.7 and 2.8

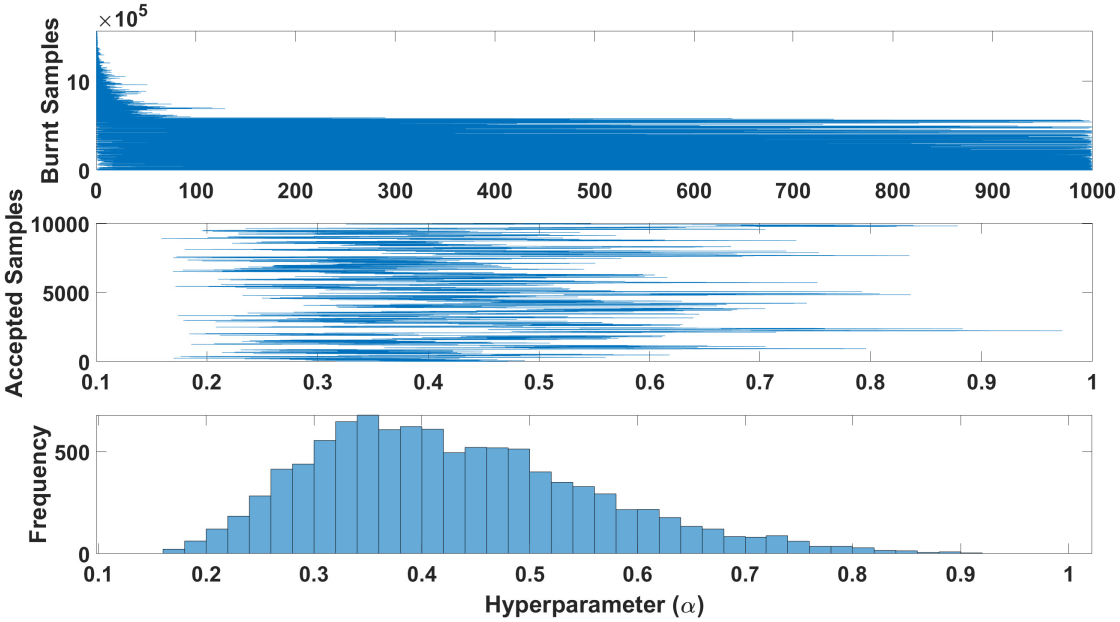


Figure 2.7: Simulated annealing results for hyperparameter α for the GP predictions shown in Figure 2.6.

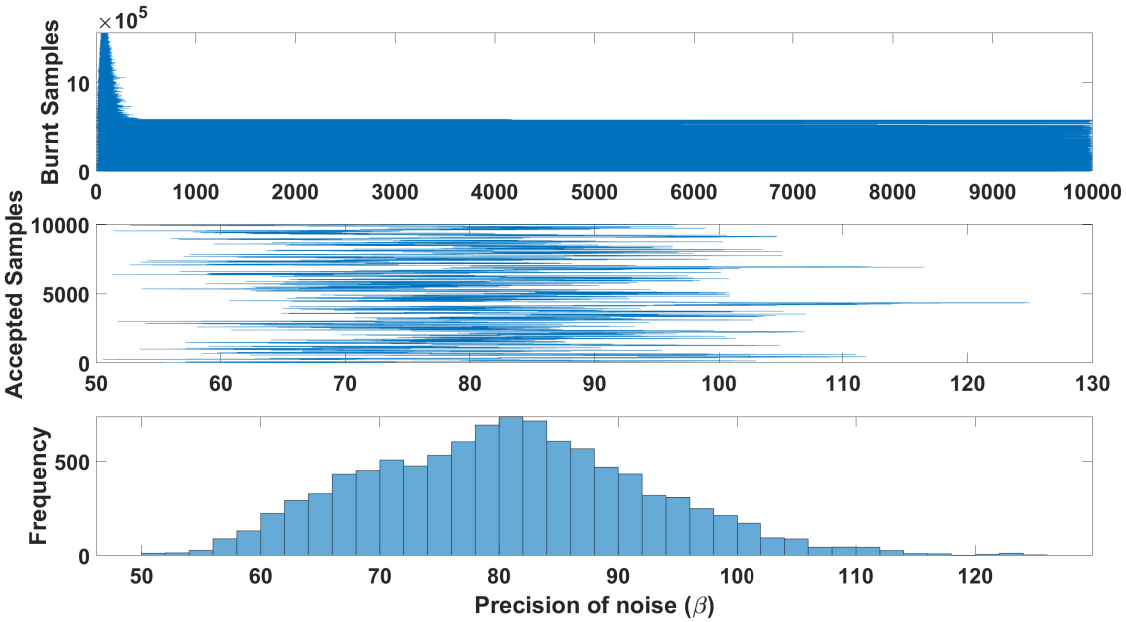


Figure 2.8: Simulated annealing results for hyperparameter β for the GP predictions shown in Figure 2.6.

2.1.4 Prediction

Here, the main aim of regression is to make predictions at previously unseen inputs that are not in the training data. Given a new input vector, $\boldsymbol{x}^*$, one can estimate the probability of observing a new point y^* given previous observations $\boldsymbol{y}$. Defining $\boldsymbol{y}^* = [y_1, y_2, \ldots, y_N, y^*]^T$, the joint distribution over $\boldsymbol{y}^*$ is

$$p(\boldsymbol{y}^*) = \mathcal{N}(\boldsymbol{y}^* \mid \boldsymbol{0}, \boldsymbol{C}_{N+1}) \tag{2.34}$$

where $\boldsymbol{C}_{N+1}$ is a $(N+1) \times (N+1)$ covariance matrix

$$\boldsymbol{C}_{N+1} = \begin{bmatrix} \boldsymbol{C}_N & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{bmatrix} \tag{2.35}$$

where

$$\boldsymbol{k} = k(\boldsymbol{x}_n, \boldsymbol{x}^*), \quad n = 1, \ldots, N \tag{2.36}$$

and

$$c = \beta^{-1} + k(\boldsymbol{x}^*, \boldsymbol{x}^*). \tag{2.37}$$

Noting the equivalence

$$\mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{C}_N & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{bmatrix}\right) \equiv \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix}\right) \tag{2.38}$$

also noting that $p(\boldsymbol{y}^*) = p(\boldsymbol{y}, y^*) \equiv p(a, b)$ the conditional Gaussian has the form

$$p(b \mid a) = \mathcal{N}\left(b \mid \boldsymbol{\mu}_{b|a}, \boldsymbol{\Sigma}_{b|a}\right) \tag{2.39}$$

where

$$\boldsymbol{\mu}_{b|a} = \boldsymbol{\mu}_b - \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}(a - \boldsymbol{\mu}_a)$$

$$\boldsymbol{\Sigma}_{b|a} = \boldsymbol{\Sigma}_{bb} - \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}\boldsymbol{\Sigma}_{ab}.$$

The conditional distribution $p(y^* \mid \boldsymbol{y})$ is a Gaussian distribution with mean and variance

$$p(y^* \mid \boldsymbol{y}) = \mathcal{N}(y^* \mid \mu, \sigma^2) \tag{2.40}$$

where μ is the mean prediction and σ^2 is the variance (which is used to measure the uncertainty in the predictions of y^*). Note that $\sigma^2 = \beta^{-1}$. Using equations (2.38) and (2.39), the mean and variance of y^* given $\boldsymbol{y}$ are [26]

$$\mu(y^* \mid \boldsymbol{y}) = \boldsymbol{k}^T\boldsymbol{C}_N^{-1}\boldsymbol{y}$$

$$\sigma^2(y^* \mid \boldsymbol{y}) = c - \boldsymbol{k}^T\boldsymbol{C}_N\boldsymbol{k}$$

which can be written more compactly using the notation

$$y^* \sim \mathbf{GP}(\boldsymbol{x}^*) \tag{2.41}$$

Figures 2.3 and 2.6 is an example of an 'ideal' fitting GP. The training data is denoted as diamonds. The GP predictions go through the centre of the points with the confidence bounds encompassing all of the training data; this is expected for an 'ideal' fitting GP. The sine data used in Figures 2.3 and 2.6 is described in equation (2.29) in which in noise model is also given. For this example, the noise model is known. Thus the correct variance σ^2 can be selected. In a real case, such as a rotorcraft, the noise model would be unknown. Therefore the optimisation technique selects the optimal hyperparameters and variance as described in Section 2.1.3.

Gaussian Process Input Structures

Depending on their input structure, GPs can be used to emulate static or dynamic relationships. The input structures shown in Section 2.1.1 can be applied to a GP. For example, the auto-regressive with exogenous inputs (ARX) structure has the format

$$\boldsymbol{x} = \begin{pmatrix} v_n \\ \\ y_{n-1} \end{pmatrix} \tag{2.42}$$

where v is a generic system input and, as before, y represents system observations, n is the time step. The ARX structure uses information from 'lagged' terms (previous observations and inputs) and the current input to help predict y_n. The difference between linear regression and GP regression is that GP is non-linear; therefore, the model is referred to as a NARX model.

The NARX model has been used as the input structure when modelling a non-linear dynamical system such as the Duffing oscillator [40, 41]. It has also been used to develop models of rotorcraft dynamics in [42, 43, 44] (the references are discussed in more detail in Chapter 3).

Two types of predictions are used in the study: one step ahead and full model predictions. These types of prediction are described here.

One Step Ahead Predictions

One step ahead predictions (OSAP) use the previously observed data to predict a single step into the future. Using **GP** to represent the prediction made by a GP (equation (2.41)) then, in our specific case, OSAP is defined as

$$y_n^* = \mathbf{GP}\left(v_n, y_{n-1}\right) \tag{2.43}$$

where y_{n-1} is the previous observation. During 'training' of the GP, OSAP are used to quantify the fidelity of the emulator. The obvious disadvantage of this approach is that the resulting model can only predict a single step into the future. Predicting further into the future requires 'full model predictions'.

Full Model Predictions

To make predictions beyond a single step, the GP model using an auto-regressive input structure must utilise previous predictions as part of the model input. The studies by Girard et al. [45], Girard [46] and Quinonero-Candela et al. [47] utilised the previous predictions as part of the model input to incorporate the uncertainty into the lagged predicted response (referred to as multi-step-ahead predictions). To illustrate the use of a previous prediction as part of the model input, consider the situation where a single prediction, at time n, has already been made according to equation (2.43). Following this, predictions of y_{n+1}^* would be realised according to

$$y_{n+1}^* = \mathbf{GP}\left(v_n, y_n^*\right). \tag{2.44}$$

The key aspect to note regarding equation (2.44) is that y_n^* (the uncertain prediction made by the GP at time n) is now part of the model input. The previous (uncertain) predictions, therefore, become inputs and carry their predictive uncertainty into future predictions [40].

To illustrate the full model predictions (FMP) used in the current thesis, a simple algorithm for generating an ensemble of predictions for FMP is shown in Algorithm 1, where R is the number of Monte Carlo samples, Y denotes a random sample and $n = 1, \ldots N$, where N is the

number of predictions. The algorithm creates R realisations one complete set of predictions N. For a more detailed algorithm for FMP, the thesis by Girard [46] is recommended.

Algorithm 1 Full Model Predictions algorithm

1: **for** $r = 1 : R$ **do**
2: $n = 1$
3: $Y^*_{(n+1),r} \sim \mathbf{GP}(\delta_{n+1}, y^*_n)$
4: **for** $n = 2 : N$ **do**
5: $Y^*_{(n+1),r} \sim \mathbf{GP}(\delta_{n+1}, Y^*_{(n),r})$
6: **end for**
7: **end for**

Duffing Oscillator Example

A Duffing oscillator is used as an example to show that the auto-regressive nature of the input structure is required to predict a dynamical system accurately. The equation of motion for the Duffing oscillator example is

$$\ddot{z} + c\dot{z} + kz + k_3 z^3 = w \tag{2.45}$$

where z is the displacement of the system, c is the linear damping, k is the linear stiffness, k_3 is the non-linear stiffness and w is a random excitation. In the case of this example, $c = 0.1$, $k = 10$ and $k_3 = 100$. Equation (2.45) was approximated using a 4th order Runge-Kutta numerical integration technique with random excitation w, a time step of 0.01 and 5000 points simulated. From the simulation, a time history of displacement was recorded, and for the GP prediction, a measurement noise with variance was added. The resulting time history of displacement is shown in Figure 2.9, where the observation is the Duffing displacement with added measurement noise, in this example case is shown as

$$\boldsymbol{t} = \boldsymbol{z} + \boldsymbol{\epsilon} \quad \epsilon_h \sim \mathcal{N}(0, \sigma) \quad h = 1, \ldots, H \tag{2.46}$$

where in this case $\boldsymbol{t}$ is the observation for the GP, $\boldsymbol{z}$ is the displacement of the Duffing oscillator, H is the number of observations, and σ is the standard deviation, which in this example is 0.005.

In the current example, an ARX input structure is used to create the GP model. For the

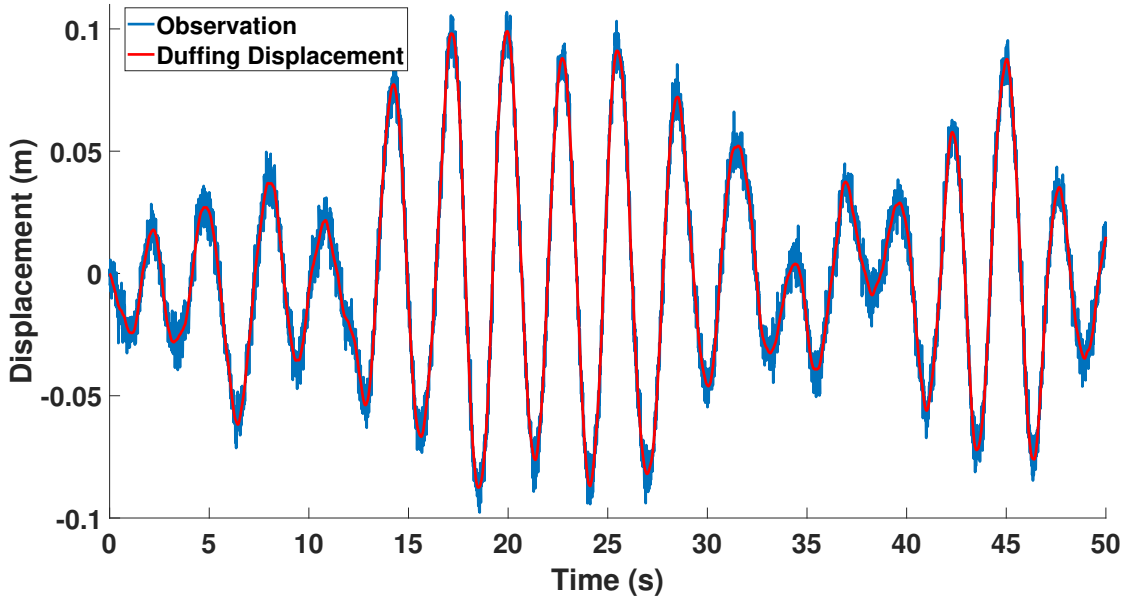


Figure 2.9: Time history of displacement for the example case Duffing oscillator.

example case of the Duffing oscillator, the ARX input structure has the form

$$\boldsymbol{x}_h = \begin{pmatrix} w_h \\ z_{h-1} \end{pmatrix} \tag{2.47}$$

where w is the excitation, z is the displacement, and h is the time step. Using the ARX input structure in equation (2.47) the GP NARX model for the Duffing oscillator is shown in Figure 2.10. The GP model was training on 50 training points and predicted 5000 points. The GP model predicts the displacement of the Duffing oscillator accurately with confidence bounds that encompass the displacement.

To justify the use of the ARX input structure for the GP to predict a dynamical system, a GP was created without the ARX aspect to investigate if the predictions are as accurate as the GP using an ARX input structure. The input structure of the GP without the auto-regressive aspect is

$$\boldsymbol{x}_h = \begin{pmatrix} w_h \end{pmatrix} \tag{2.48}$$

where only the current excitation is used for the input structure. The GP predictions for the

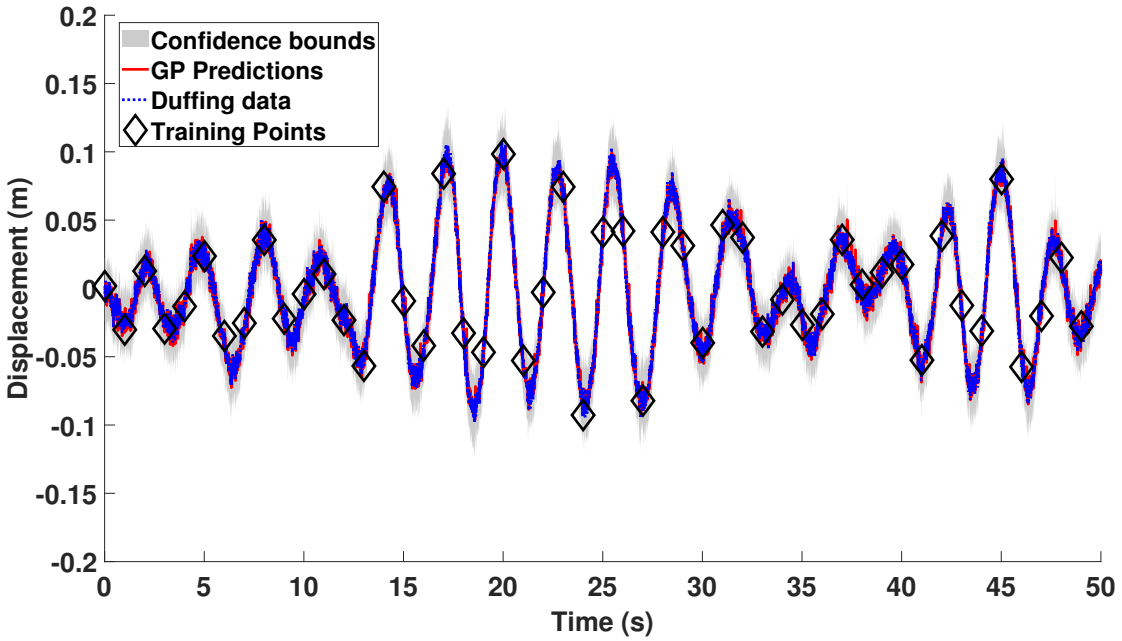


Figure 2.10: GP predictions for the Duffing oscillator response described in equation (2.45) using the ARX structure shown in equation (2.47).

non-ARX input structure is shown in Figure 2.11. The same Duffing oscillator data is used in Figure 2.11, trained on 50 training points and predicted 5000. The GP predictions are very poor, thus demonstrating the need for an ARX input structure to predict a dynamical system accurately. Using this knowledge, it could also be the case for the prediction of the rotorcraft dynamics, which is investigated in later chapters.

2.1.5 Choice of Kernel

The kernel chosen for the current study was used by Higdon et al. [48]. The kernel takes the form of

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \prod_{i}^{N_D} \alpha^{(4(\boldsymbol{x}_n^i - \boldsymbol{x}_m^i)^2)} \tag{2.49}$$

where x_i is used to represent the ith element of vector $\boldsymbol{x}$ and N_D is the number of dimension of the input vector. The chosen kernel from equation (2.49) allows the hyperparameter to always be between zero and one. The hyperparameter bounds are beneficial when using a property of GPs that determines the relevance of inputs, as the values of the bounded hyperparameters enable ease of understanding (which input is relevant or not). The relevance property of the GP

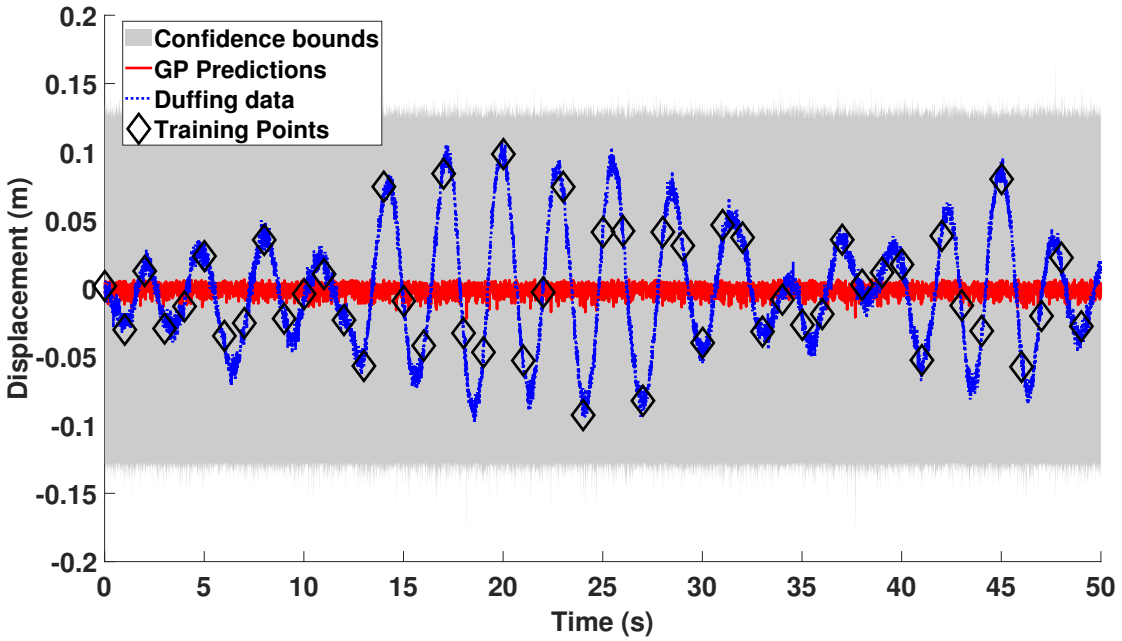


Figure 2.11: GP predictions for the Duffing oscillator response described in equation (2.45) without an ARX input structure applied, the input structure shown in equation (2.48) is used.

is referred to as automatic relevance determination (ARD); this is discussed in more detail in Section 2.1.6. As discussed in Section 2.1.3, the proposal width can be a problem when running the MCMC optimisation process. The choice of this kernel bounds the hyperparameter between zero and one, thus bounding the proposal width, therefore, eliminating the problem of searching an infinite target distribution. Figure 2.12 displays the effect of the hyperparameter on the kernel function in equation (2.49). When the α value is close to 1 the functions are smoother than a value that is smaller such as 0.1, however, when the value of α is 1 the function has no smoothness.

2.1.6 Automatic Relevance Determination (ARD)

Automatic relevance determination (ARD) is a process which aims to establish which inputs are most 'relevant'; in this case, the relevance is determined by the dependence of inputs to the GP model. For example, if the most 'relevant' input were removed, this would have a detrimental effect on the GP predictions. However, if the least 'relevant' input were removed, the GP model would still produce accurate predictions.

To enable the use of ARD in GP regression, for the example case of the Duffing oscillator, a

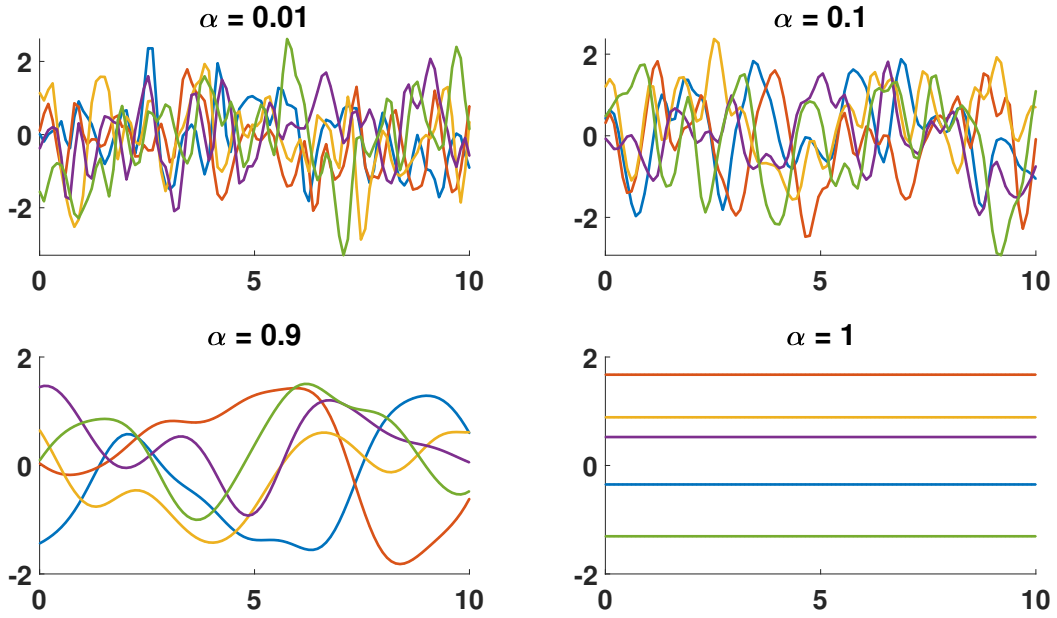


Figure 2.12: Examples of how the hyperparameter α affects the kernel function shown in equation (2.49).

hyperparameter (α) is assigned to each input such that the kernel function is written as

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \alpha_1^{4(w_n^x - w_m^x)^2} \times \alpha_2^{4(z_{n-1} - z_{m-1})^2} \tag{2.50}$$

The hyperparameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ can be used to compute the relevance of the inputs. The relevance of the input is judged on the corresponding hyperparameter value, hence why it is beneficial to use the kernel in equation (2.49), as the hyperparameters are bounded between 0 and 1. A hyperparameter value of 1 suggests the corresponding input is not relevant; therefore, a value close to 0 indicates a high relevance. Optimisation of the hyperparameters provides evidence as to which inputs are needed to predict the response of the system.

The relevance determination can help decrease the complexity of the model, as well as its computational cost, by removing the inputs that are not relevant. To estimate these hyperparameters, an optimisation technique needs to be conducted, which has already been discussed.

Propagating Hyperparameter Uncertainty

When creating a GP model, it is not clear which hyperparameter produce the best results; therefore, an optimisation process such as MCMC is used to select the hyperparameter. The

hyperparameters selected from MCMC are usually chosen by the mean of the stationary distribution or the most probable hyperparameters in the stationary distribution. It is unclear how much effect the hyperparameter selection will have on the GP predictions. It is therefore beneficial to visualise this effect, which is called hyperparameter uncertainty. The hyperparameter uncertainty method shown in the current section requires hyperparameter ($\boldsymbol{\theta} = [\alpha, \beta]$) samples from MCMC. The GP also has uncertainty associated with its predictions, which can be combined with the hyperparameter uncertainty. Including both sources of uncertainty, the uncertainty in the GPs predictions is written as

$$y^* \sim \mathcal{N}(\mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta})) \tag{2.51}$$

where $\sigma^2 = \beta^{-1}$, MCMC estimates the hyperparameters:

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta} \mid \boldsymbol{y}). \tag{2.52}$$

Which is Ancestral sampling, where samples are taken from equation (2.52) and also taken from equation (2.51). Ancestral sampling is when one draws samples from a given probability distribution, and a sample from another probability distribution governs the given probability distribution. In the case of hyperparameter uncertainty, the first sample ($\boldsymbol{\theta}$) is drawn from $p(\boldsymbol{\theta} \mid \boldsymbol{y})$, this hyperparameter sample is then used to govern the distribution, $\mathcal{N}(\mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta}))$, in which a sample y^* is drawn. The posterior samples are generated using MCMC. This method is used by Bishop [26]. An example of the hyperparameter uncertainty is shown in Figure 2.13, which uses the same Duffing oscillator data displayed in Section 2.1.4. The GP results in Figure 2.13 is trained on 50 training points, predicts 5000 and uses a different value for α and β for each realisation[1], as shown in equation (2.52). The number of realisations in Figure 2.13 is 10000. The α and β values are selected from the accepted samples of the stationary distribution produce by the simulated annealing algorithm. The accepted samples of the stationary distribution for the Duffing oscillator GP predictions using an ARX input structure (equation (2.47)) is shown in Figure 2.14 in the form of a histogram. Previously in

[1] A realisation is defined as one complete set of GP predictions.

Figure 2.10, the mean of the stationary distribution accepted samples is used for the selection of the hyperparameters.

It is clear from Figure 2.13 that the hyperparameter uncertainty does affect the GP. Hyperparameter uncertainty related to rotorcraft simulator model development is investigated in Chapter 4.

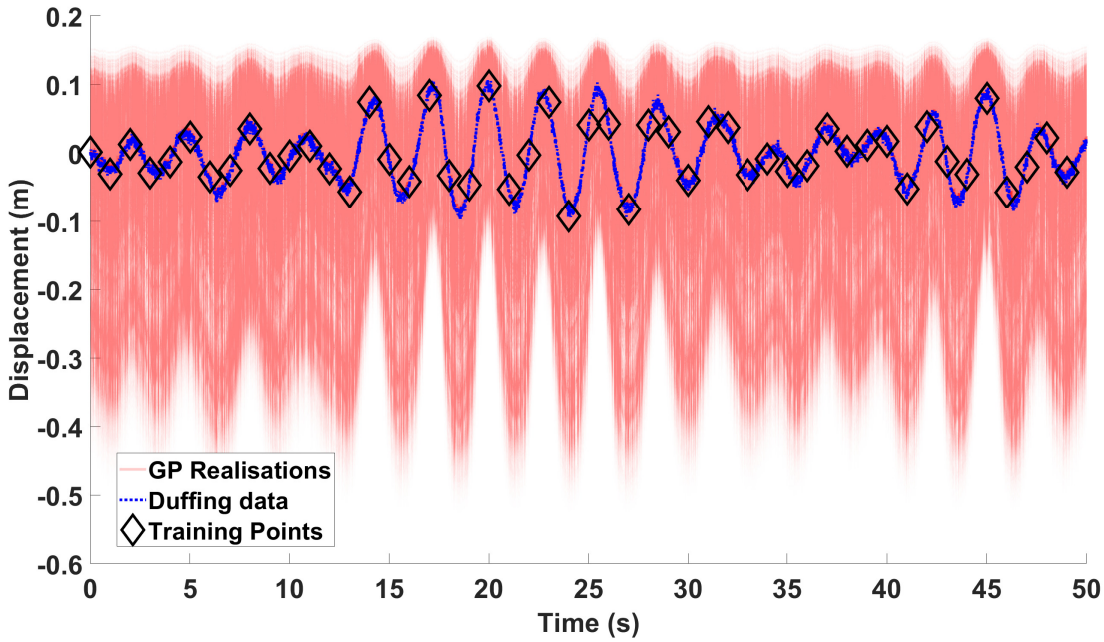


Figure 2.13: GP predictions for the Duffing oscillator response described in equation (2.45) with ARX structure applied, the input structure is shown in equation (2.48) is used, with hyperparameter uncertainty incorporated.

2.2 Summary

The current chapter showed the theory for the GP regression, optimisation processes, the choice of kernel, the GP input structure and the incorporation of hyperparameter uncertainty. Two different types of predictions are discussed in detail; one step ahead and full model predictions. One step ahead prediction can only predict one step in the future while the FMP approach can be used to realise predictions at an arbitrary point in the future. The outcomes from the theory shown are that for a dynamical system, in the case of the current chapter, a Duffing oscillator, an ARX input structure is required. In the subsequent chapters, the theory shown in the current chapter will be used to predict the dynamics of a rotorcraft, which may also need an ARX input structure. The effect of hyperparameter uncertainty on the rotorcraft dynamics will

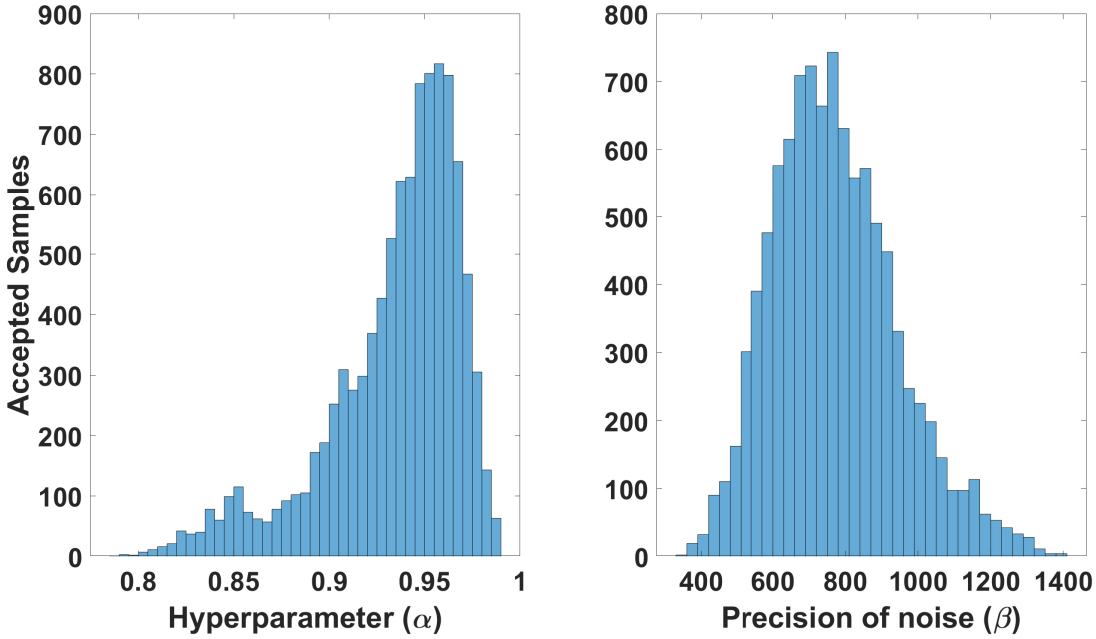


Figure 2.14: Histogram of the accepted samples of the simulated annealing results for the GP predictions using an ARX input structure for hyperparameters α and β.

be investigated in Chapter 4, as the hyperparameter uncertainty affected the GP realisations of the Duffing oscillator.

Chapter 3

Introduction to the Rotorcraft Dataset

In the current chapter, the Bo105 rotorcraft and the datasets used to create the machine learning model are introduced. The Bo105 rotorcraft datasets are explained in detail, and the corresponding physical law based (FLIGHTLAB) models are discussed as this is used as a comparison to the machine learning models. Different input structures for the machine learning models are investigated, and these are described at the end of the chapter.

3.1 Method

Two sets of comparison data were used in the current thesis. The first was obtained from flight test data to judge the efficacy of the machine learning models to predict complex rotorcraft dynamic responses. While the second was from the output of a FLIGHTLAB physics based model. Both were relating to the response of a Bo105 rotorcraft.

3.1.1 Bo105 Rotorcraft

The Bo105 is a twin-engined rotorcraft, in the 2.5 ton-class, which has fulfilled a number of roles in transport, offshore, police and military missions. For the generation of a physical rotorcraft simulation program, all characteristics of the rotorcraft have to be known by their physical value. For vehicle rigid-body modelling, the data needed consist of the rotorcraft parameters, such as main rotor data, tail rotor data, fuselage and empennage aerodynamics and flight control system data [49]. The Bo105 rotorcraft parameters used to generate the model in this study

was first extracted from the book by Padfield [50] which itself incorporates open-source data collected from GARTEUR HC(AG06) activities in [51]. The flight control system data were first taken from the report by Heffley et al. [52]. The complete database that was provided to the GARTEUR Action Group HC(AG16)[53] was then utilised.

FLIGHTLAB

FLIGHTLAB is commercial modelling software that can be used for the construction and application of vehicle simulation models. The software is provided by Advanced Rotorcraft Technology (ART), and as such, is tailored towards the rotorcraft market [54]. 'The objective of FLIGHTLAB is to promote Concurrent Engineering (work methodology for performing tasks in parallel) by providing a simulation tool capable of multidisciplinary support with selective fidelity modelling options' [55]. Models are developed using a library of physically-based modelling components [54]. FLIGHTLAB uses a modular approach to simulation model building at a level appropriate to the simulation data available (so, e.g., an engine could be a simple power/thrust against throttle position look-up table or it could be a more detailed thermodynamic representation where individual components such as crankshafts and gearboxes are modelled). The philosophy is that individual engineering disciplines within a project can attend to the components of a model for which they are responsible. All of the individual components can then be brought together under a common simulation framework. FLIGHTLAB capabilities are provided in the FLIGHTLAB documentation [55], and a summary version is provided in the thesis by Lawrence [56]. The multi-body approach to rotorcraft modelling was adopted for the Bo105 due to the nature of the data available for it, as described in the University of Liverpool deliverable [49].

In addition to the flexibility afforded by FLIGHTLAB, the simulation engineer can further choose to model a vehicle using a variety of techniques, primarily:

1. Full vehicle model data. FLIGHTLAB can process multi-variable datasets that represent the dynamics of the total simulated vehicle. This method is akin to how an industrial

aerospace company might store its model datasets.

2. Multi-body modelling. Using this method, individual airframe and system components are connected to form larger sub-systems of the vehicle. The complete model is then built up from the interconnected sub-systems [51].

The FLIGHTLAB model shown in the thesis was implemented and configured by an 'expert' simulation engineer from the University of Liverpool's flight science and technology research group. The model parameters, as described by a deliverable by Aristotel [49], were constructed from the generic template provided by Advanced Rotorcraft Technology (ART). The model consists of 44 states (every component with degrees of freedom will have states [55]), 18 translational and rotational body states, 4 propulsion states and 22 rotor states, incorporating flap and lead-lag rotation for each individual rotor blade.

3.1.2 Flight Test Data

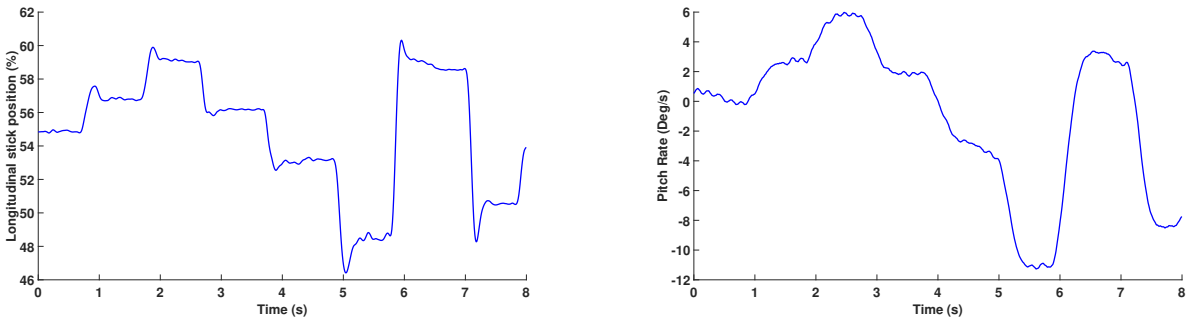
The flight test database used for this study is the so-called 'AGARD database', which was used by the AGARD Working Group FMP WG 18, on 'Rotorcraft System Identification'. The dataset was delivered to the University of Liverpool as part of the GARTEUR HC(AG-16) Rotorcraft Pilot Couplings [57]. The dataset had three different manoeuvres; the doublet, the 3-2-1-1 and the sweep, which were all undertaken during a steady-state horizontal flight at 80kt and a density altitude of above 3000ft standard atmosphere. The manoeuvre chosen for the work presented in the thesis was the 3-2-1-1; this is due to the manoeuvre exciting elements of different pilot input frequencies. The 3-2-1-1 is more complex manoeuvre than the doublet but not as complicated as the frequency sweep; therefore, it was decided that 3-2-1-1 would be a good 'test' of the machine learning models. The 3-2-1-1 dataset used in the thesis consisted of four manoeuvres, which were the 3-2-1-1 manoeuvre for each input. The 3-2-1-1 is a multistep input, which is a series of alternative steps[58]. Each step is a multiple of a user-defined time unit; for example, the first step would be a multiple of three, hence the name 3-2-1-1. The input has a broad frequency spectrum which is advantageous for modelling. The inputs were longitudinal stick position, lateral stick position, pedal position and collective lever where the on-axis responses are the pitch rate, roll rate, yaw rate, and heave respectively. The datasets

and manoeuvres are shown in Table 3.1. Table 3.1 also displays the corresponding figures for each manoeuvre. The Bo105 rotorcraft investigated in the current thesis is treated as having four inputs; longitudinal stick position (δ^x), lateral stick position (δ^y), pedal position (δ^p) and collective lever (δ^o). The outputs investigated correspond to the rotorcraft's four axes: pitch rate (p), roll rate (q), yaw rate (r) and heave ($\dot{h}$).

Dataset	Manoeuvre	On-axis response	Figure
1	3-2-1-1 Longitudinal stick position (δ^x)	Pitch rate (p)	3.1
2	3-2-1-1 Lateral stick position (δ^y)	Roll rate (q)	3.2
3	3-2-1-1 Pedal position (δ^p)	Yaw rate (r)	3.3
4	3-2-1-1 Collective lever (δ^o)	Heave ($\dot{h}$)	3.4

Table 3.1: The datasets corresponding to the manoeuvres used in the thesis.

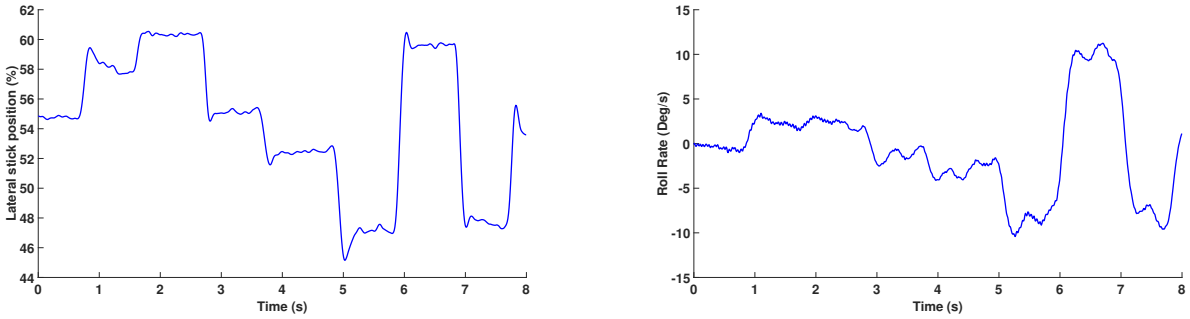
The manoeuvre shown in Figure 3.1 is referred to as the 3-2-1-1 longitudinal stick position. Figure 3.1a is the 3-2-1-1 excited longitudinal stick position, and Figure 3.1b is the on-axis pitch rate response corresponding to the longitudinal stick position.



(a) Longitudinal stick position. (b) Pitch rate response.

Figure 3.1: Bo105 flight test data of the 3-2-1-1 Longitudinal manoeuvre with the on-axis response of pitch rate.

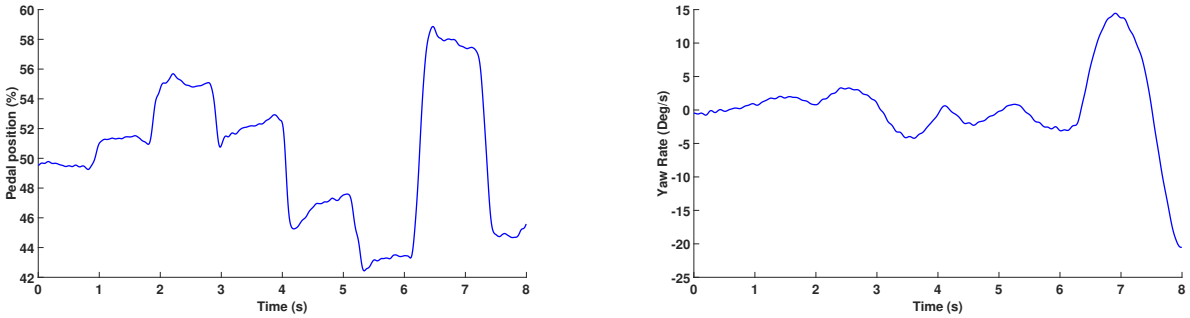
Figure 3.2a displays the 3-2-1-1 excited lateral stick position, and Figure 3.2b shows the on-axis roll rate response corresponding to the lateral stick position. The manoeuvre shown in Figure 3.2 is referred to as the 3-2-1-1 lateral stick position.



(a) Lateral stick position. (b) Roll rate response.

Figure 3.2: Bo105 flight test data of the 3-2-1-1 Lateral manoeuvre with the on-axis response of roll rate.

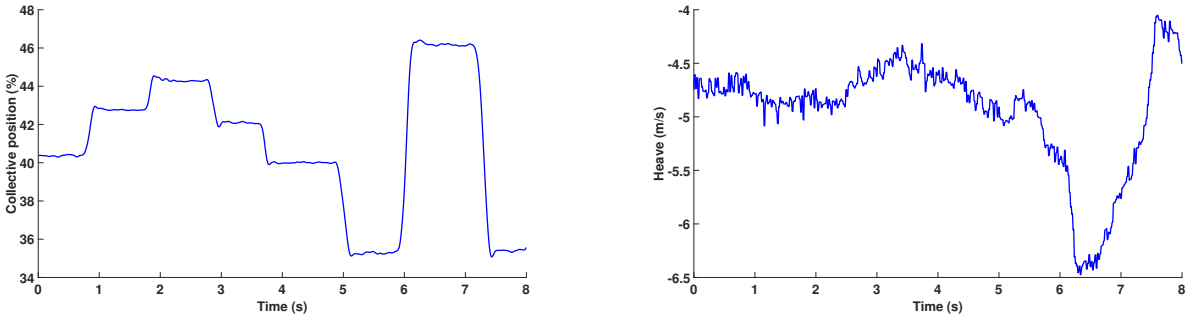
Figure 3.3a shows the 3-2-1-1 excited pedal position, and Figure 3.3b displays the on-axis yaw rate response corresponding to the pedal position. The manoeuvre shown in Figure 3.3 is referred to as the 3-2-1-1 pedal position.



(a) Pedal position. (b) Yaw rate response.

Figure 3.3: Bo105 flight test data of the 3-2-1-1 Pedal manoeuvre with the on-axis response of yaw rate.

Figure 3.4a shows the 3-2-1-1 excited collective position, and Figure 3.4b displays the on-axis heave response corresponding to the collective position. The manoeuvre shown in Figure 3.4 is referred to as the 3-2-1-1 collective position.



(a) Collective position. (b) Heave response.

Figure 3.4: Bo105 flight test data of the 3-2-1-1 Collective manoeuvre with the on-axis response of heave.

3.1.3 Regression Models Applied to the Bo105 Data

The previous chapter used a Duffing oscillator to illustrate that when attempting the predict a dynamical system, the auto-regressive aspect of the input structure is required to predict a Duffing oscillator accurately. In the literature, a NARX model has been used in [43, 44] for neural network models of rotorcraft dynamics and in [42] for a support vector regression model of rotorcraft dynamics. The two approaches use a black-box modelling approach to identify the dynamics of a rotorcraft. The difference between the given approaches and the GP is that the GP approach can describe the confidence in its predictions. GPs have been used on unmanned aerial vehicles (UAVs) to capture the dynamics of the UAVs [59] and for path planning [60]. In the case of the current thesis, the dynamic system is a Bo105 rotorcraft. Generally speaking, to create an encompassing solution, it is beneficial to investigate all the input structures shown in Section 2.1.1. The input structures for on-axis pitch rate predictions are displayed in Table 3.2, where y is the relevant observation (pitch rate) and n is a time step.

Input structure	**Format**
Moving average (MA)	$\boldsymbol{x}_n = \left(\delta^x_{n-1}\right)$
Auto-regressive (AR)	$\boldsymbol{x}_n = \left(y_{n-1}\right)$
Auto-regressive moving average (ARMA)	$\boldsymbol{x}_n = \left(\delta^x_{n-1}, y_{n-1}\right)$
Auto-regressive with exogenous inputs (ARX)	$\boldsymbol{x}_n = \left(\delta^x_{n}, y_{n-1}\right)$
Auto-regressive moving average with exogenous inputs (ARMAX)	$\boldsymbol{x}_n = \left(\delta^x_{n}, \delta^x_{n-1}, y_{n-1}\right)$

Table 3.2: Input structures applied to the Bo105 rotorcraft data.

The machine learning regression models presented in the thesis aim to create a more accurate

initial model for the flight simulator. A more accurate initial model would potentially reduce time and cost for the many iterations required to make an accurate flight simulator model. The datasets used in the current work are for only one manoeuvre; however, it displays evidence that the machine learning models can create a more accurate initial model when compared to the initial physic based FLIGHTLAB model. Ideally, the approach could be used to model a full manoeuvre in the flight simulator; however, this would require a large amount of data, Chapter 7 outlines a method that has the potential to reduce the number of training points for a machine learning model. Chapter 8 displays results using the approach described in Chapter 7 and demonstrates the scalability to larger datasets. Testing the models on a full flight manoeuvre is not demonstrated in the current work but could be used in future projects.

The implementation of the GP models into the flight simulator is not demonstrated in the current thesis. The implementation of the GP could potentially be undertaken using the following methods:

- Embedding the GP 'black-box' model into the code of the simulator to drive the responses of the flight simulator.

- Using the GP model to correct the physical law based FLIGHTLAB model, which is referred to as the 'grey-box' model (this method is discussed in more detail in Chapter 9.

- Utilising both the physical law based FLIGHTLAB model and the GP model to use one as a primary model and the other a back up. Once the primary model deviates far from the truth, the back up model can step in. For example, if the GP model is the primary model and the predictions become uncertain (which can be shown be the confidence in the predictions), the response of the flight simulator would be driven by the physical law based model, until the GP model is 'more certain' with the predictions.

The implementation of the GP models into the flight simulator is discussed in the more detail in Section 10.2.

3.2 Summary

In the current chapter, the flight test datasets of the Bo105 rotorcraft used to create the machine learning model displayed in the current thesis are introduced. The corresponding FLIGHTLAB models are described which are used for comparison to the machine learning models. The input structures used in the subsequent chapters for the machine learning model are defined using the Bo105 rotorcraft flight test data.

Chapter 4

Feasibility Study: Bo105 Data

In the previous Chapter, 3, the Bo105 rotorcraft was introduced. The data from the Bo105 is used in the current and subsequent Chapters (5, 6, 8 and 9) to create regression models. In this chapter, the main aim is to investigate the feasibility of using a data-based regression model to predict the response of the pitch rate (on-axis) and roll rate (off-axis) using a 3-2-1-1 longitudinal cyclic input. Four manoeuvres were discussed in the previous chapter; however, to check the feasibility of using a data-based regression model to predict the response of the pitch and roll rate, the flight test data of Bo105 rotorcraft performing the 3-2-1-1 manoeuvre with longitudinal control input is used in this chapter. Firstly, linear regression models using different input structures are investigated, followed by GP models. Each model examines the one step ahead and full model predictions (discussed in detail in Section 2.1.4). Alongside the feasibility of the GP regression models, the effect of hyperparameter uncertainty (background in Section 2.1.6) is explored. The results shown in the current chapter is a preliminary step which directed the research in the subsequent chapters.

The results produced in the current chapter and subsequent chapters were created from code implemented by the author using the algorithms shown in Chapter 2. The coded algorithms, which include, GP implementations and optimisation techniques, were written without libraries/packages and executed in Matlab.

In this chapter, the flight test data of Bo105 rotorcraft performing the 3-2-1-1 manoeuvre

with longitudinal control input is used, which is referred to as the 3-2-1-1 longitudinal input manoeuvre. The 3-2-1-1 longitudinal control input of the Bo105 rotorcraft is shown in Figure 4.1, which displays a time series of the displacement of the longitudinal stick position (regression model training points displayed as black diamonds).

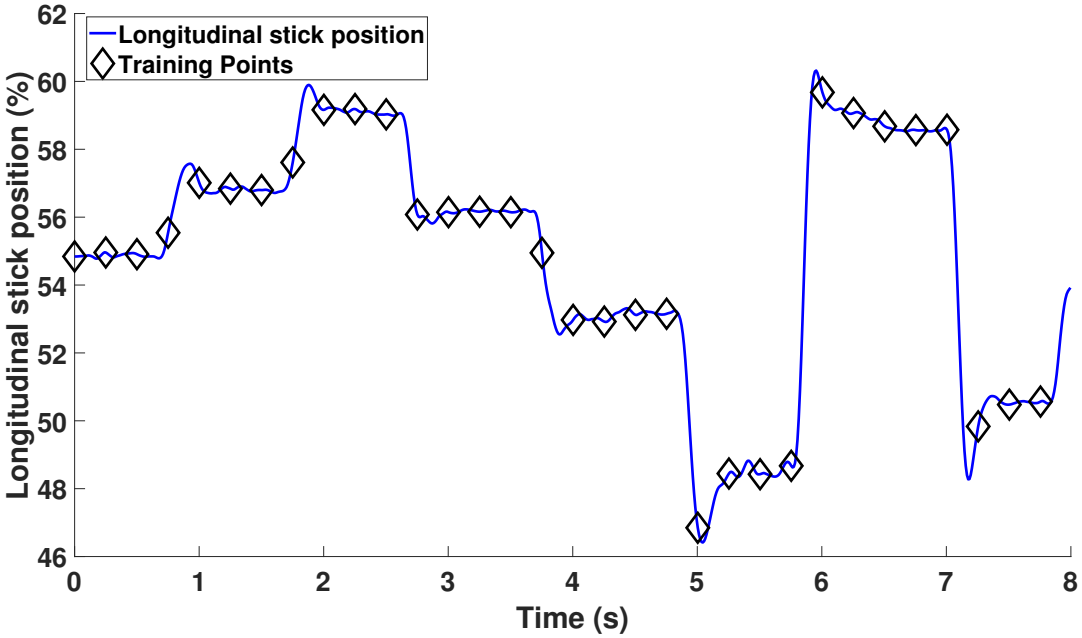


Figure 4.1: 3-2-1-1 longitudinal control input

Throughout the current chapter and the rest of the thesis, the regression model predictions are compared to the Bo105 flight test data. It is, therefore, necessary to quantify this error for comparison. The function used to measure the similarity between the regression model predictions, and the Bo105 flight test data is [33]:

$$J = \frac{100}{N\sigma_y^2}\sum_{n=1}^{N}(y_n - R(\boldsymbol{x}_n))^2 \tag{4.1}$$

where y_n represent the nth point in the time history of the Bo105 flight test data, σ_y^2 is the variance of the Bo105 flight test data, $\boldsymbol{x}_n$ is the input into the regression model with an example shown in equation (4.7) and R is the denotation of the regression model predictions. The similarity function is such that if the regression model predictions produced the mean of the Bo105 flight test data (denoted $\overline{y}$) the similarity function returns a value of 100:

$$J = \frac{100}{N\sigma_y^2}\sum_{n=1}^{N}(y_n - \overline{y})^2 = 100. \tag{4.2}$$

Different levels of fidelity (closeness to the real rotorcraft) would be acceptable depending on the role that the flight simulator is utilised for; for example rotorcraft development, qualification or training. It therefore difficult to assess the performance of model without establishing the intended role of the flight simulator. In the current thesis, the similarity measure is used to assess the performance of GP models to the physical law based model.

4.1 Linear Regression Models

Firstly, to check the feasibility of applying regression models to the Bo105 rotorcraft, a linear regression model defined in Section 2.1.1 is applied. In the current section, different input structures for linear regression models are investigated for the prediction of the pitch rate response. All linear regression models shown in the current section use 32 training points. The reason the regression model was trained on 32 points was to provide hold-out experiments for validation, and assess how the model performs on data that it has not 'seen' before.

4.1.1 Moving Average (MA)

Firstly, the moving average (MA) input structure, as described in Section 2.1.1, is investigated. The moving average input structure is a linear combination of past inputs. In the case of the Bo105 rotorcraft, the on-axis input for pitch rate response is the longitudinal stick position (δ^x):

$$\boldsymbol{x}_n = \begin{pmatrix} \delta^x_{n-1} \end{pmatrix} \tag{4.3}$$

As discussed in Section 2.1.4, the most straightforward test of a model is the one step ahead predictions (OSAP). The pitch rate OSAP from the linear regression model using the moving average input structure is displayed in Figure 4.2. The similarity measure between the flight test data and the OSAP is 47.87. As this is the most straightforward test of the model, and it has performed poorly on the OSAP, it will not be investigated further. For the MA input structure, it is not possible to make full model predictions (FMP) as they require the input

structure to contain a previous prediction.

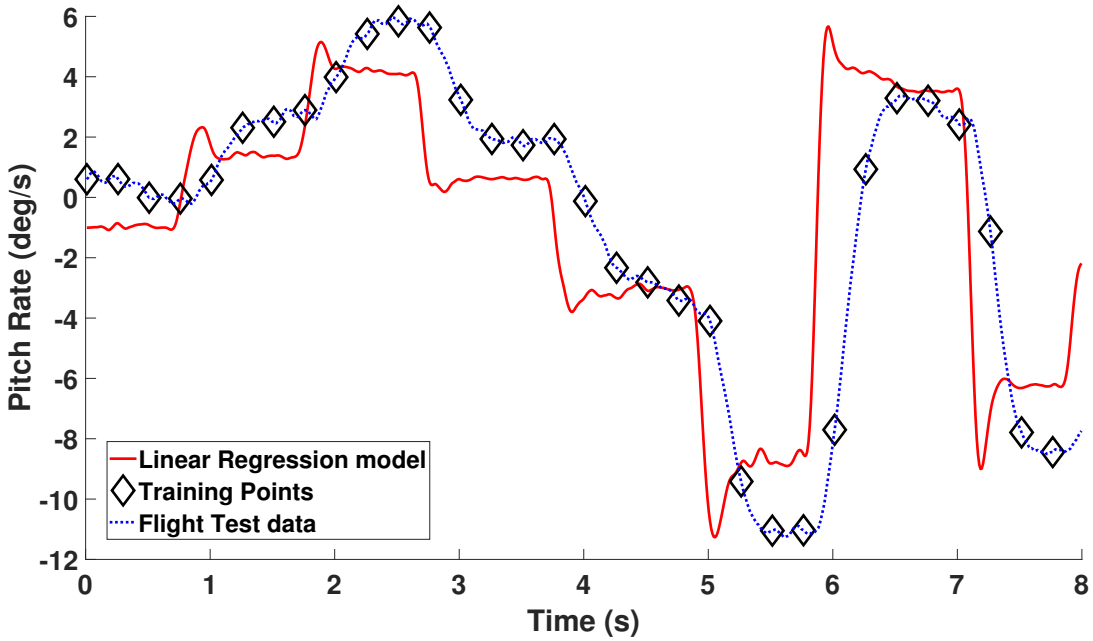


Figure 4.2: One step ahead pitch rate predictions with a moving average input structure.

4.1.2 Auto-regressive (AR)

The auto-regressive input structure contains lagged previous outputs; in the case of the Bo105 rotorcraft, this corresponds to the previous pitch rate. The input structure has the form

$$\boldsymbol{x}_n = \begin{pmatrix} y_{n-1} \end{pmatrix} \tag{4.4}$$

The OSAP for the linear regression model using the auto-regressive input structure, predicting the pitch rate response is displayed in Figure 4.3. The similarity measure between OSAP and the flight test data is 0.05; the predictions using the AR input structure are more accurate than the MA input structure.

The FMP for the pitch rate predictions from the linear regression model using the AR input structure is displayed in Figure 4.4. It can be seen that the predictions struggle to match the pitch rate dynamics of the rotorcraft, and there is a similarity measure of 107.19 between the pitch FMP and the flight test data. It can therefore be assumed that the AR input structure should not be used for the regression model.

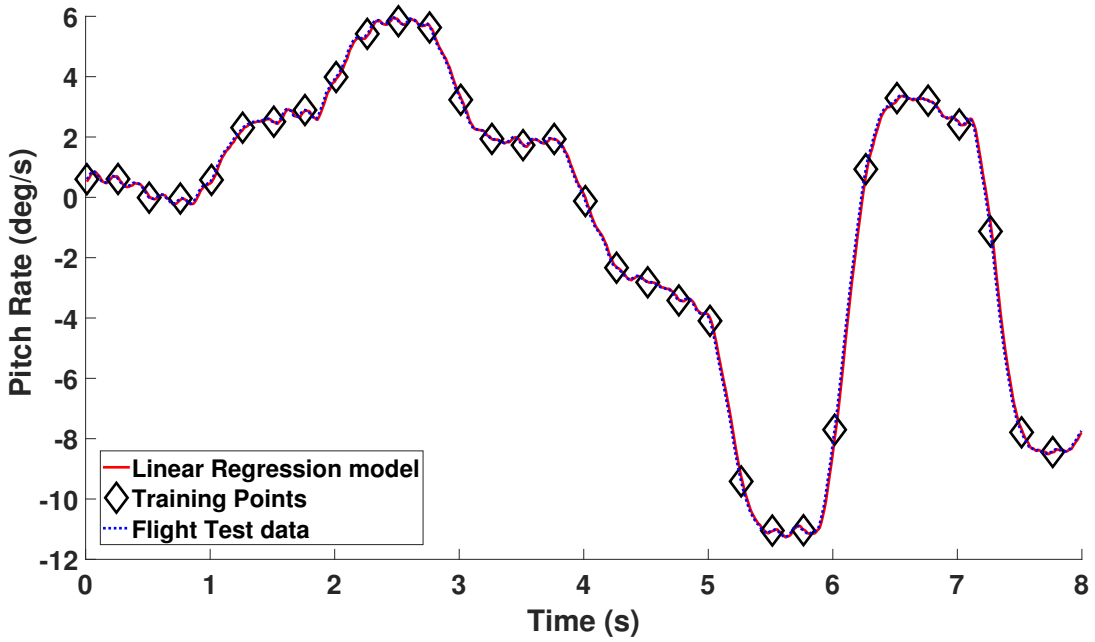


Figure 4.3: One step ahead pitch rate predictions with an auto-regressive input structure.

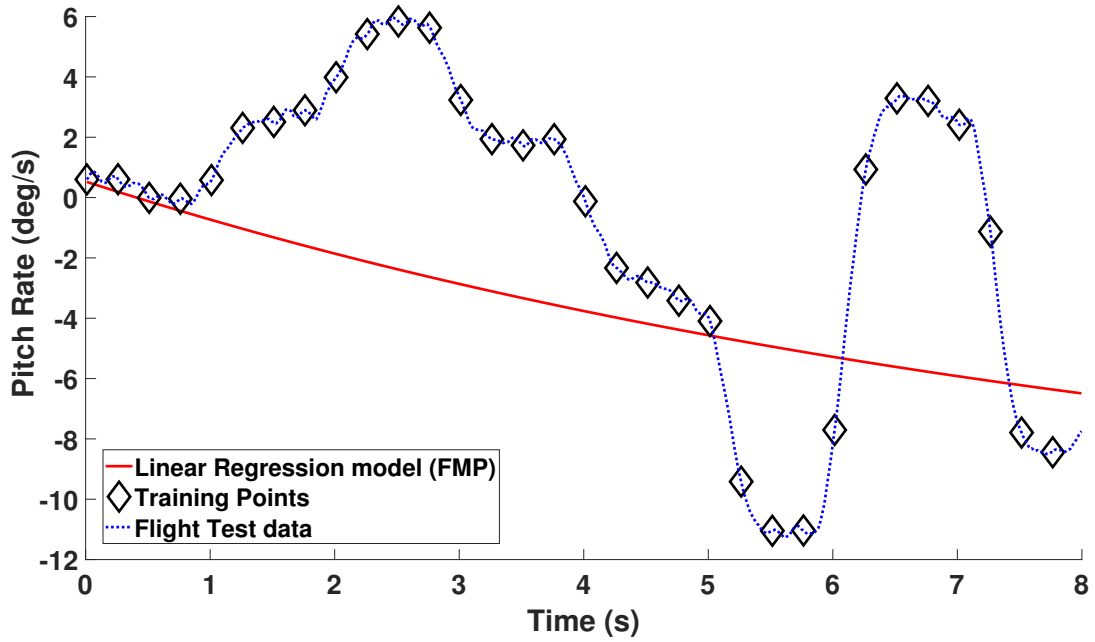


Figure 4.4: Full model pitch rate predictions with an auto-regressive input structure.

4.1.3 Auto-regressive with Exogenous Inputs (ARX)

The auto-regressive with exogenous inputs (ARX) input structure builds on the AR input structure by including the current input. In this case, the current input is the longitudinal stick position. The ARX structure uses the input and output of the Bo105 rotorcraft, thus enabling the model to use information from both inputs and outputs. The ARX input structure has the format:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \\ y_{n-1} \end{pmatrix} \tag{4.5}$$

The pitch rate OSAP from the linear regression model using an ARX input structure is shown in Figure 4.5. It can be seen that the OSAP are very accurate; this is expected as the OSAP for the AR input structure were also very accurate. The similarity measure between OSAP and flight test data is 0.03. However, as previously discussed the FMP are the best test of the model.

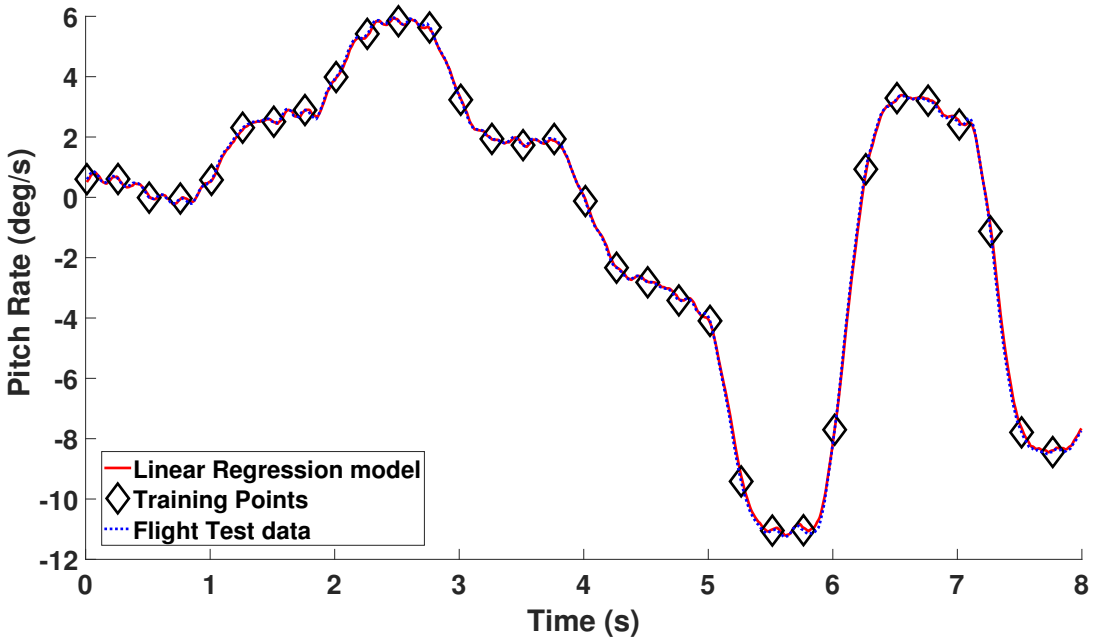


Figure 4.5: One step ahead pitch rate predictions with an auto-regressive with exogenous input structure.

The FMP for the pitch rate response from the linear regression model using ARX input (shown

in equation (4.5)) is shown in Figure 4.6. It can be seen that the FMP are more accurate than the FMP in Figure 4.4, providing evidence that the current inputs improve the accuracy of predictions. However, the predictions do miss the dynamics of the Bo105 pitch rate response at around 5 to 6 seconds and between 7.5 and 8 seconds. The similarity measure between the flight test data and the FMP using the ARX input structure is 14.66. Potential improvements to the model include changing the input structure (which is investigated next) and using other regression models, including non-linear models (which is investigated in subsequent sections in the current chapter).

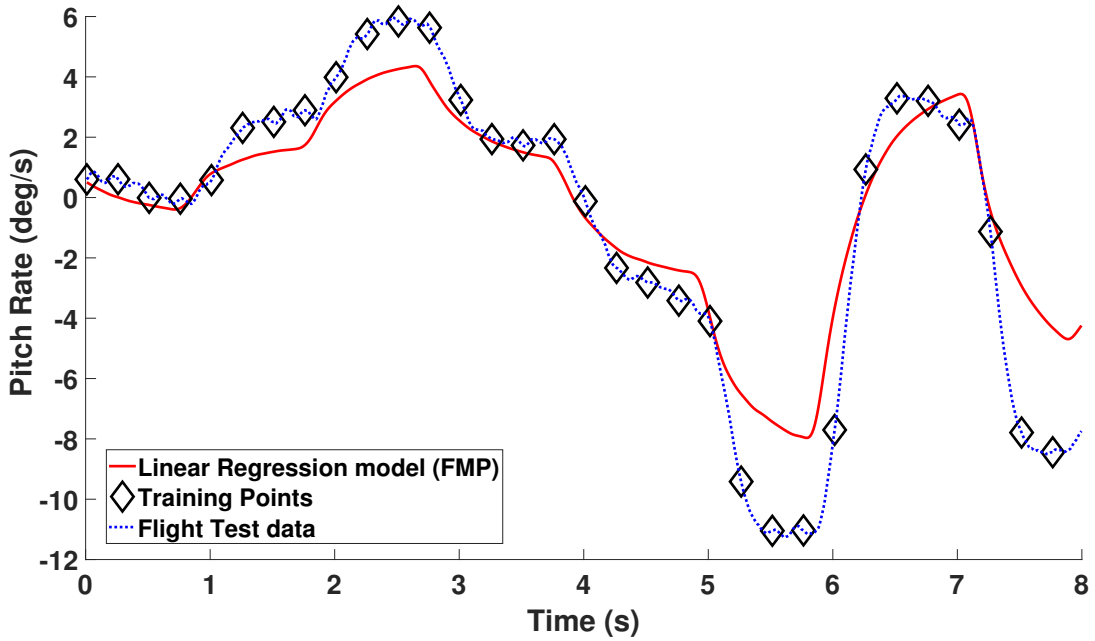


Figure 4.6: Full model pitch rate predictions with an auto-regressive with exogenous input structure.

4.1.4 Auto-regressive Moving Average (ARMA)

Combining two of the previously discussed input structures, AR and MA, produces an auto-regressive moving average (ARMA) model. Application to the Bo105 rotorcraft uses a lagged pitch rate response and a lagged longitudinal stick position, such that

$$\boldsymbol{x}_n = \begin{pmatrix} y^x_{n-1} \\ \delta^x_{n-1} \end{pmatrix} \tag{4.6}$$

The OSAP for pitch rate response from the linear regression model utilising the ARMA input structure (equation (4.6)) is shown in Figure 4.7. The similarity measure is 0.03, which is the same as the linear regression model utilising the ARX input structure, which is feasible as both input structures contain a lagged pitch rate response and a pilot input (longitudinal stick position).

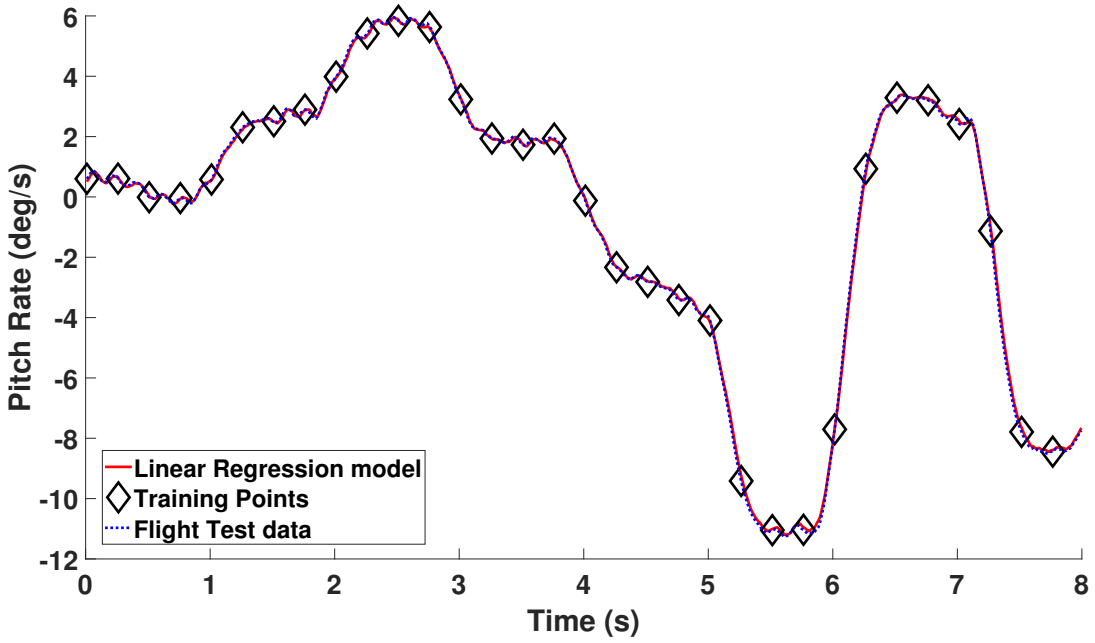


Figure 4.7: One step ahead pitch rate predictions with an auto-regressive moving average input structure.

The FMP for the pitch rate response from the linear regression model utilising the ARMA input structure is displayed in Figure 4.8. The similarity measure for the pitch rate FMP from the ARMA linear regression model, compared to the flight test data is 14.57. The FMP for the linear regression using ARX and ARMA input structures are very similar, and both produce more accurate predictions than the linear regression models utilising the MA and AR input structures. The FMP from the ARMA linear regression model misses the same pitch rate dynamics as the ARX models (towards the latter end of the predictions).

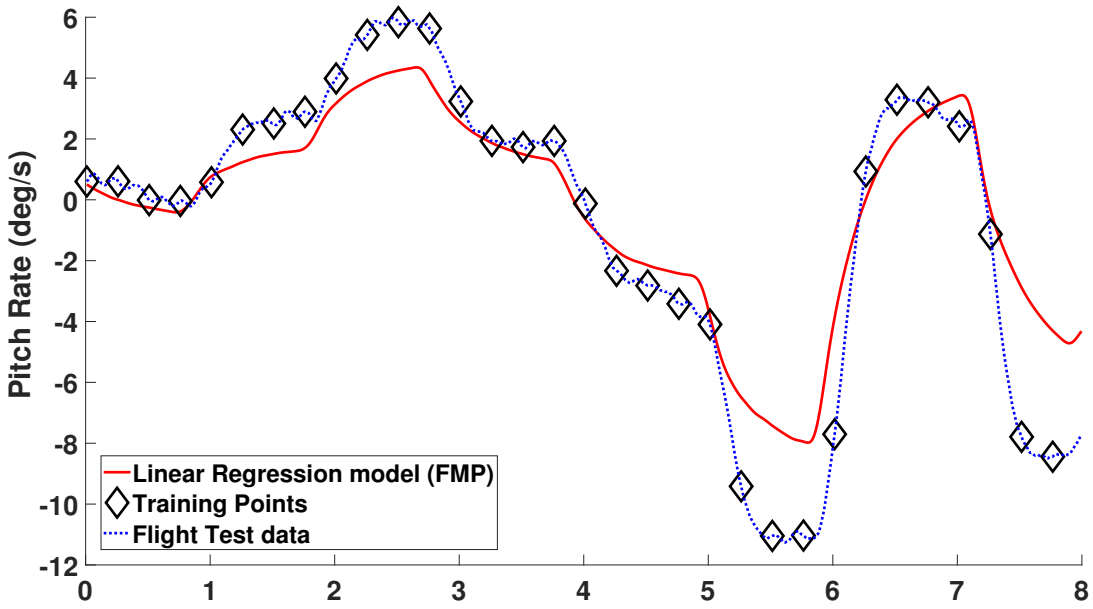


Figure 4.8: Full model pitch rate predictions with an auto-regressive moving average input structure.

4.1.5 Auto-regressive Moving Average with Exogenous Inputs (ARMAX)

Finally, combining the ARX and MA input structures creates the auto-regressive moving average with exogenous inputs (ARMAX) input structure. In the context of the Bo105 rotorcraft, the ARMAX input structure contains the current and one lagged longitudinal stick position with one lagged pitch rate response, such that

$$\boldsymbol{x}_n = \begin{pmatrix} \delta^x_n \\ \delta^x_{n-1} \\ y_{n-1} \end{pmatrix} \tag{4.7}$$

Figure 4.9 displays the OSAP for pitch rate response from the linear regression model utilising the ARMAX input structure. The similarity measure between the pitch rate OSAP and the flight test data is 0.03, which is the same as the ARX and ARMA linear regression models.

The pitch rate full model predictions from the linear regression model using the ARMAX input structure is shown in Figure 4.10. The similarity measure between the flight test data and the

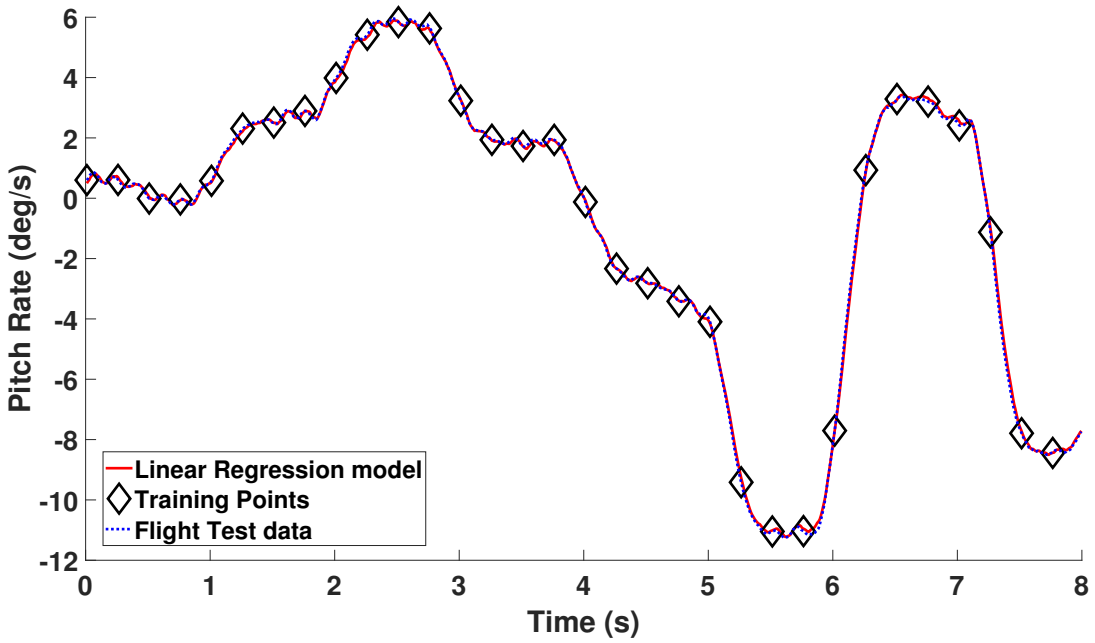


Figure 4.9: One step ahead pitch rate predictions with an auto-regressive moving average with exogenous input structure.

FMP is 22.00; this is worse than the ARX and ARMA linear regression models, it could be due to using the extra pilot input, that has added extra 'noise' to the model. Another possible hypothesis for the ARMAX input structure producing less accurate predictions than the ARX and ARMA models is that the ARMAX model is overfitting, i.e., finding a correlation between inputs that holds over the training data but does not generalise beyond it.

4.1.6 Similarity Measure

Table 4.1 displays the similarity measures for the predictions from the linear regression models that utilised five different input structures. It can be seen that in terms of the FMP accuracy, the linear regressions models that used the input structures of ARX and ARMA produced the most accurate results. This result provides evidence that both inputs and outputs of the Bo105 rotorcraft are required to capture the dynamics of the pitch rate response accurately. As the results for the ARX and ARMA are very similar, it was chosen to continue the investigation using the input structure of ARX as this input structure is consistent with others used in the literature (see studies by Kumar et al. [43] and Omkar et al. [44]).

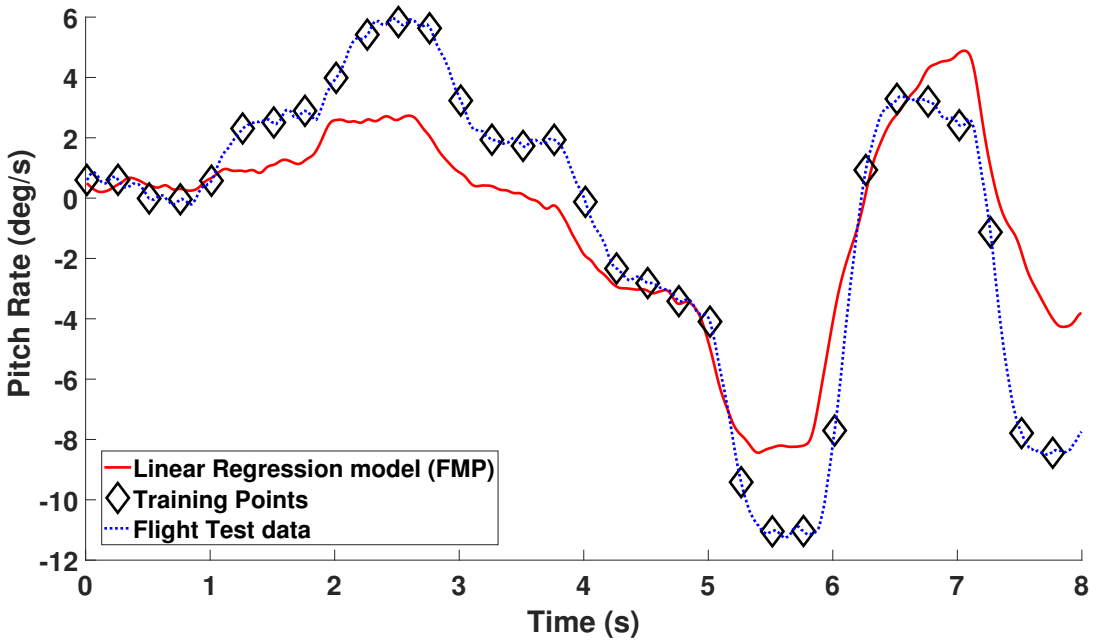


Figure 4.10: Full model pitch rate predictions with an auto-regressive moving average with exogenous input structure.

Prediction	Input Structure	Prediction	Similarity measure
Pitch rate	Moving Average (MA)	OSA	47.87
	Auto-regressive (AR)	OSA	0.05
		FMP	107.19
	Auto-regressive with exogenous inputs (ARX)	OSA	0.03
		FMP	14.66
	Auto-regressive moving average (ARMA)	OSA	0.03
		FMP	14.57
	Auto-regressive moving average with exogenous inputs (ARMAX)	OSA	0.03
		FMP	22.00

Table 4.1: The similarity measures for the pitch rate predictions from linear regression models using different input structures.

4.2 Configurations for the Gaussian Process Models

In the current section, a GP model utilising the ARX input structure is investigated to determine if the GP model can produce more accurate predictions than the linear regression models displayed in the previous section. Secondly, to test if the GP model can produce accurate predictions of the response of the off-axis response of the longitudinal stick position, which is the roll rate response.

Table 4.2 shows the different GP configurations used in the current chapter. The configurations are referred to as 'cases'.

Manoeuvre	**Prediction**	**Input structure**	**Case**
3-2-1-1 Longitudinal input	Pitch rate	Longitudinal input with one lagged pitch rate term	1
	Roll rate	Longitudinal input with one lagged roll rate term	2

Table 4.2: The input and hyperparameter structures used for the GP model displayed in Chapter 4.

The column denoted 'inputs structure' in Table 4.2 is the input for the GP; for example, in case 1 on pitch rate prediction, the ARX input structure is the current longitudinal pilot input with one lagged pitch rate. To illustrate this, the ARX input structure that is used in case 1 is:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \\ y_{n-1} \end{pmatrix} \tag{4.8}$$

where n is the time step, $\boldsymbol{x}$ is the input of the GP, δ_n^x is the current longitudinal stick position, y_{n-1} in this example is the pitch rate. Note that the input to the GP is different from the input of the pilot of the Bo105 rotorcraft. In the current chapter, one hyperparameter is used; multiple hyperparameters are investigated in later chapters. For more detail on the use of hyperparameters, see the section on kernel structure (Section 2.1.5).

4.3 Gaussian Process Results

In this section, the two cases shown in Table 4.2 are investigated to explore if GP models using an ARX input structure can produce accurate predictions for the pitch and roll rate. Each response is investigated individually, and two GP models are created using the GP theory discussed in Section 2.1.2 and applied to the cases shown in Table 4.2. The longitudinal cyclic Bo105 input, shown in Figure 4.1, is used to excite the rotorcraft axes. The on-axis pitch rate is a direct response to the longitudinal cyclic input. Due to the nature of the rotorcraft, there is cross-coupling in practically every axis-pairing [3]. However, Padfield [3] states that the major types of coupling in a rotorcraft are the pitch to roll, roll to pitch, collective to pitch and collective to yaw. Therefore, in the current section, the on-axis pitch rate and off-axis roll rate responses using the longitudinal cyclic input are investigated. If there is a small amount of cross-coupling in the yaw rate and heave axes, there will be little correlation between the longitudinal input and the yaw rate and heave. For the interested reader, the results for the yaw rate and heave using the longitudinal cyclic input are, however, shown in Appendix A.

Note that, here, the GP models do not use all of the available training data, which was 800 points. The GPs only use 32 training points, denoted by diamonds in Figure 4.1. The GP is trained (more detail on GP training is shown in Section 2.1.2) on 32 points, while the predictions interpolate between the training points by using all the available data (more information on the GP prediction, see Section 2.1.4). As previously stated, the reason the GP was trained on 32 points was to reduce the time complexity of the optimisation process and to provide hold-out experiments for validation. In this section, the training points are selected with equal time intervals (every 25th point of the full dataset). The training points used in the current chapter may not be the optimal solution a method for automatically selecting the training points is discussed in Chapter 7. The GP models in this section (pitch and roll rate) are compared to the FLIGHTLAB predictions (more detail on FLIGHTLAB data is given in Chapter 3) and the Bo105 flight test data.

4.3.1 Pitch Rate - On-axis

Firstly the pitch rate response was investigated, in which the GP used the ARX input structure of the current longitudinal input with one lagged pitch rate term utilising one hyperparameter (case 1, Table 4.2), as shown in equation (4.8). As previously mentioned, the GP used 32 training points, which are denoted by diamonds in Figure 4.1. For the initial GP OSAP and the FMP (without hyperparameter uncertainty) the hyperparameters were located by the simulated annealing algorithm (Section 2.1.3), where the mean of the final accepted samples was used to select the hyperparameters. To reach the final accepted samples, 'burn-in' is required, which is shown as Monte Carlo trace plots in Figure 4.11. For both hyperparameters (α and β) during the 'burn-in' the MCMC samples converge. The final accepted samples for the simulated annealing results are shown in Figure 4.12. The hyperparameters calculated for the pitch rate OSAP and FMP (without hyperparameter uncertainty) were 0.984 and 73174, for α and β respectively. Firstly, the mean of the accepted samples is used to investigate if the results produce accurate predictions. Later in the section, hyperparameter uncertainty is explored.

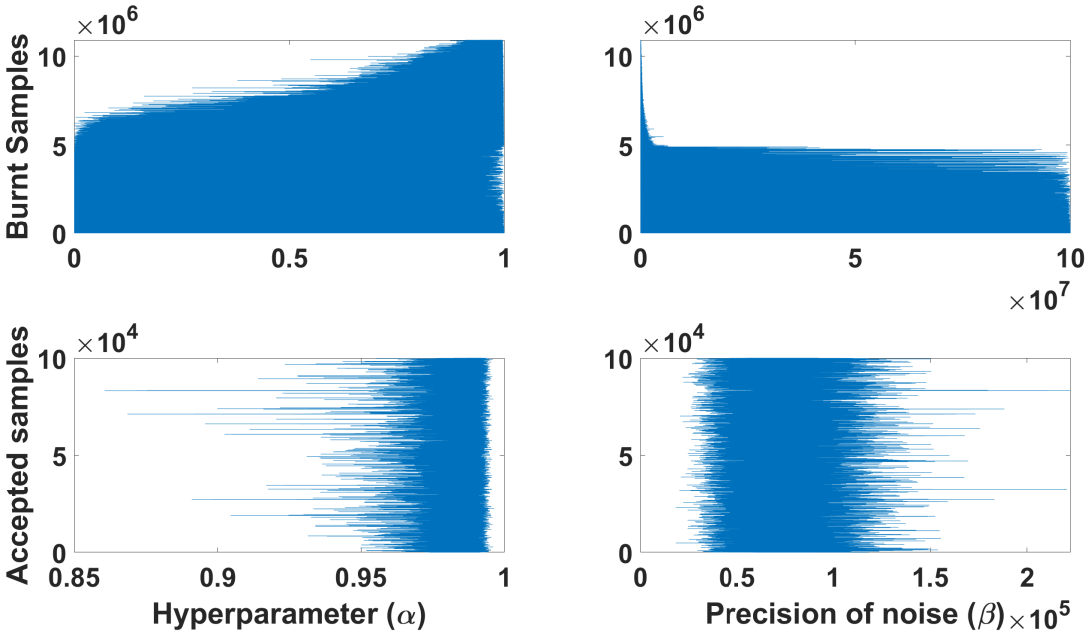


Figure 4.11: Simulated annealing burn-in and accepted samples used to create the pitch rate GP model using the input structure containing the current longitudinal pilot input and one lagged pitch rate term (case 1, Table 4.2).

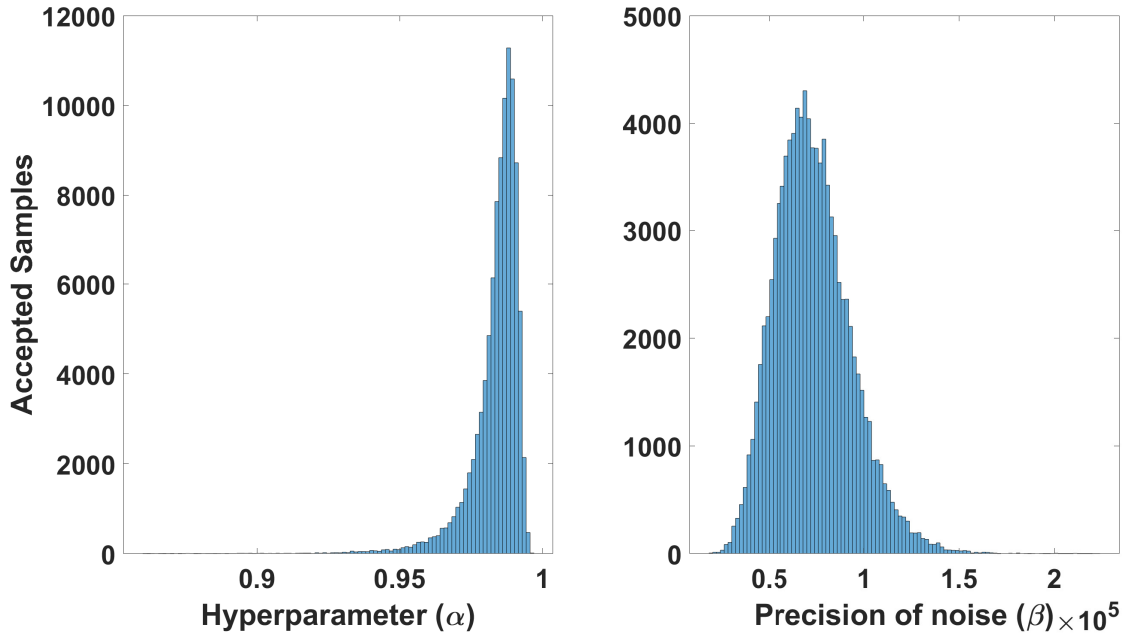


Figure 4.12: Simulated annealing results used to create the pitch rate GP model using the input structure containing the current longitudinal pilot input and one lagged pitch rate term (case 1, Table 4.2).

Figure 4.13 shows the OSAP made by the GP. As was discussed previously (Section 2.1.4), it would be expected that the predictions are very close to the Bo105 flight test data. The legend in Figure 4.13 states that a grey shaded area shows the confidence bounds (3 standard deviations from the mean). The confidence bounds are displayed in Figure 4.13; however, the bounds are minimal. The similarity measure using the function shown in equation (4.1) calculated across all points and not just the training data, is 0.02.

Figure 4.14 shows the pitch rate GP FMP realisations for the ARX input structure (more detail on FMP is given in Section 2.1.4) containing the current longitudinal pilot input with one lagged pitch rate term (case 1, Table 4.2). Note that, in Figure 4.14, the Monte Carlo realisations from the FMP are shown, alongside the Bo105 flight test data and the FLIGHTLAB predictions. The FMP realisations use 10,000 Monte Carlo samples and plotted using semi-transparent lines, such that darker regions represent areas where more realisations were observed[1].

Figure 4.14 displays two sets of FMP realisations; the first set has no hyperparameter uncertainty shown as red realisations, which use the mean of the final accepted simulated annealing

[1]10,000 Monte Carlo samples are used for all the FMP realisations shown in the current thesis.

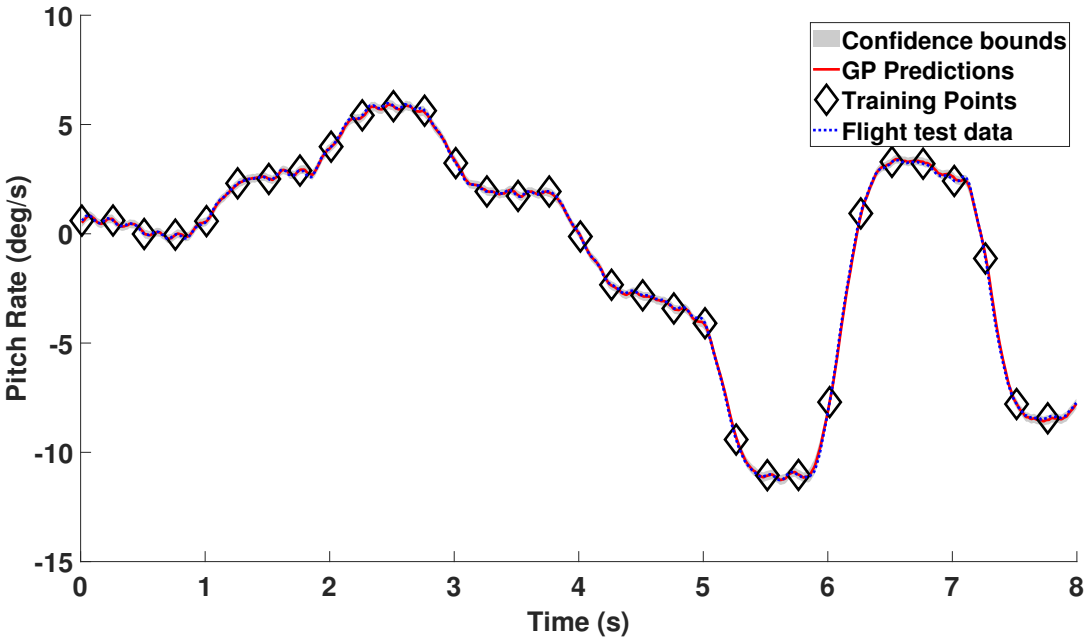


Figure 4.13: GP one step ahead pitch rate predictions using the input structure containing the current longitudinal pilot input and one lagged pitch rate term utilising one hyperparameter (α) (case 1, Table 4.2).

samples. The second has hyperparameter uncertainty (more detail in Section 2.1.6), which are displayed as green realisations. The overlapping realisations are grey. The comparison is investigated, as one does not know how much effect the hyperparameters uncertainty will have on the GP predictions. The hyperparameter uncertainty was incorporated by using the hyperparameter randomly selected from the simulated annealing for each realisation of the FMP (more information is provided in the hyperparameter uncertainty Section, 2.1.6). The FMP predictions in Figure 4.14 show little difference between the two GP realisations (hyperparameter uncertainty and no hyperparameter uncertainty), this implies that the hyperparameter uncertainty does not have a significant effect on the FMP realisations.

The FMP is a better test of the GP model, as due to the ARX input structure, as shown in equation (4.5), the lagged output is a prediction from the GP. For this model, the previous prediction becomes an input to the next prediction and, hence, predictive uncertainty is carried through the simulation. It can be seen in Figure 4.14 that the realisations do not encompass all of the flight test data; this is discussed in more detail at the end of the chapter.

The comparison of the flight test data (blue line), FLIGHTLAB (black dashed line) and the

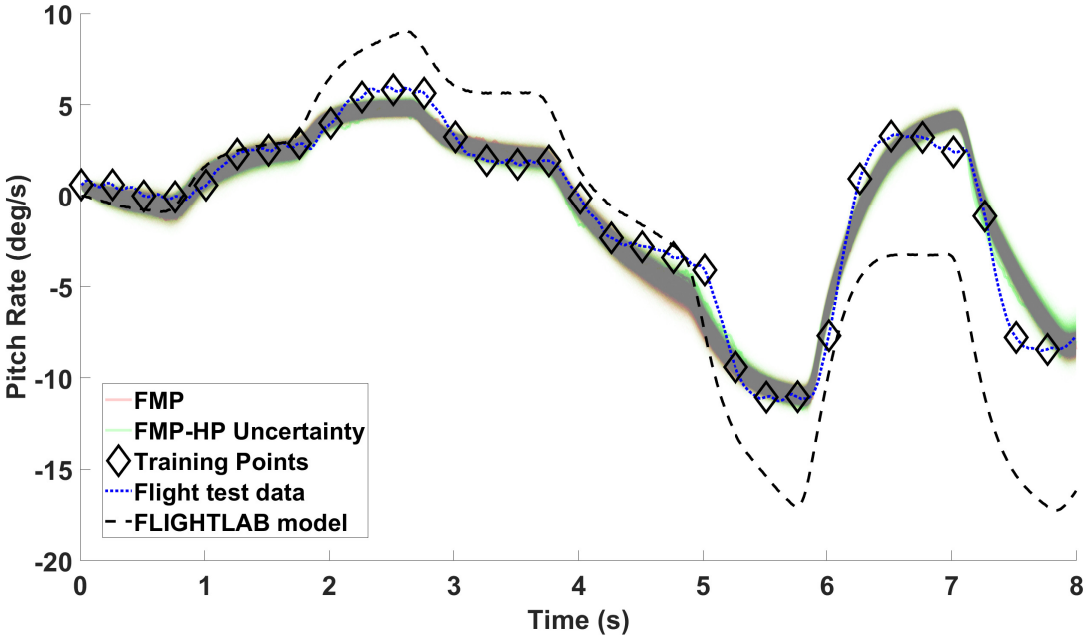


Figure 4.14: Realisations of the GP pitch rate full model prediction using the input structure containing the current longitudinal pilot input and one lagged pitch rate term utilising one hyperparameter (α) (case 1, Table 4.2), with a comparison to the corresponding FLIGHTLAB model.

GP FMP realisations are shown in Figure 4.14. The GP models and FLIGHTLAB capture the essence of the real aircraft response. However, it is apparent that the GP models outperform the FLIGHTLAB model in this case; in general, the GP predictions remain much closer to the flight test data throughout the manoeuvre, and particularly in its latter stages. To try to quantify this improvement, the similarity measure using equation (4.1), between the mean of the FMP realisations (no hyperparameter uncertainty) and flight test data for GP FMP is 5.25 and for the FLIGHTLAB model is it 75.02, as previously mentioned a value less than five provides a good correlation and less than one is considered excellent [33]. These confirm what is easily observable in the figures; that, for the primary axis response, the GP model outperforms the FLIGHTLAB model. The similarity measure (shown in Section 4.1.6) between the mean GP FMP realisations with hyperparameter uncertainty and the flight test data is 5.10.

Figure 4.15 compares the FMP for the linear regression model and a GP (GP) model using the ARX input structure (shown in equation(4.5)). A red line displays the FMP for the linear regression model, and the GP FMP are shown by green realisations. Alongside the linear regression and GP models, the flight test data (blue dashed line) and the FLIGHTLAB model

(black dashed line) are displayed in Figure 4.15. The results shown in Figure 4.15 provide evidence that the GP creates more accurate predictions than the linear regression model using the ARX input structure. The outperformance of the GP compared to the linear regression model is also confirmed when using the similarity measure, the similarity measure between the mean of the GP FMP realisations and the flight test data is 5.25. The similarity measure between the linear regression FMP and the flight test data is 14.66. The linear models displayed in Section 4.1 provide an excellent basis to build on and highlight which input structures provide the most accurate pitch rate predictions. In the rest of the thesis, a non-linear GP model is investigated.

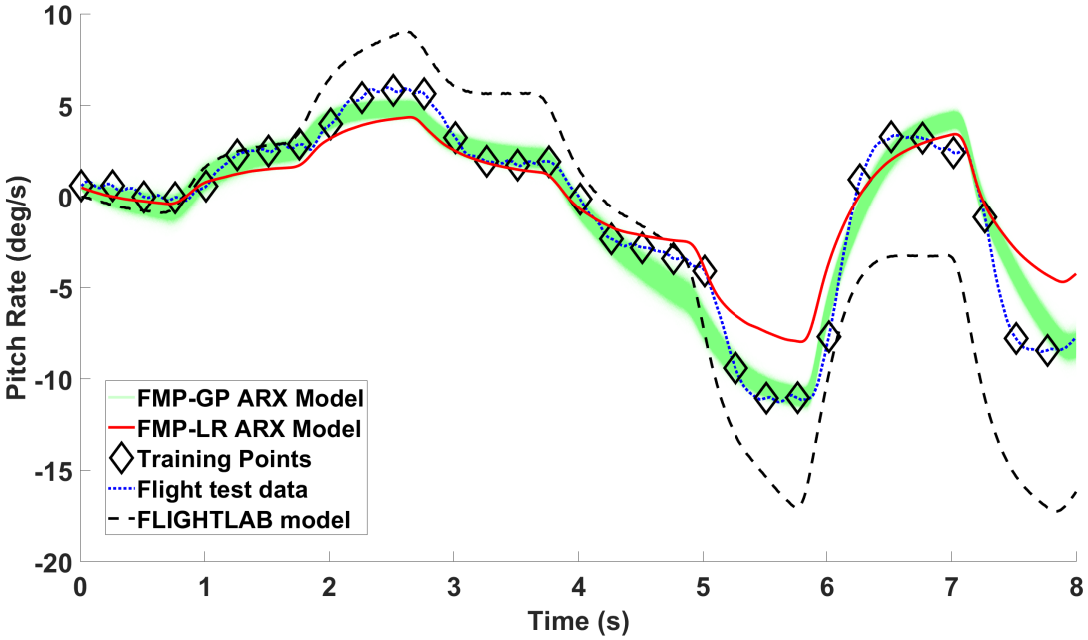


Figure 4.15: Full model pitch rate predictions with an auto-regressive moving average with exogenous input structure.

For reference sake, Figure 4.16 displays two sets of FMP for the pitch rate prediction using the ARX input structure containing the current longitudinal pilot input with one lagged pitch rate term. The two sets of FMP are from two GP models, the first GP model was trained on 400 data points, and the second GP model was trained on 32 data points. Note that, the full dataset had 800 data points. The red realisations display the FMP for the GP model trained on 400 data points, the FMP for the GP model trained on 32 data points is displayed by green realisations, and for the sake of the clarity of the visualisation, the training points

are not plotted. Comparing the two sets of realisations on Figure 4.16 that there is very little difference, providing evidence that using 32 training points produced predictions that are of comparable accuracy as a model trained on a significantly larger training set. The accuracy of the model using 32 training points is confirmed when comparing the similarity measures; the similarity measure between the mean GP FMP realisations using 400 training points and the flight test data is 4.70, compared to 5.25 for the GP trained on 32 training points. It can therefore be concluded that the extra training points are not adding a large amount of new information to the GP model.

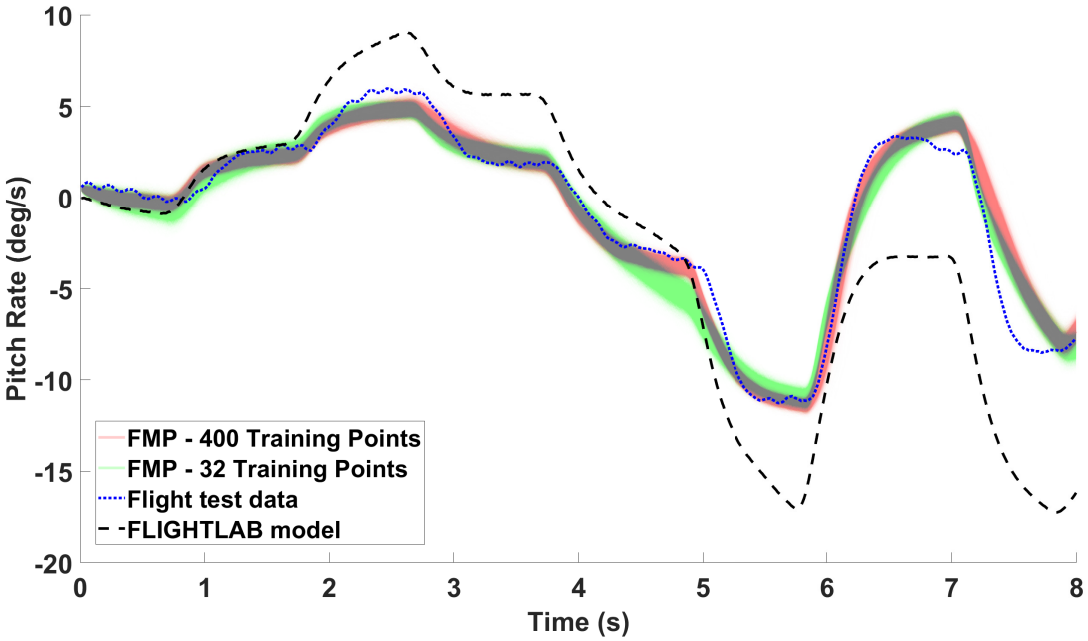


Figure 4.16: Two sets of realisations of the GP pitch rate full model prediction from two GP models using the ARX input structure containing the current longitudinal pilot input and one lagged pitch rate term utilising one hyperparameter trained on 400 training points and 32 training points (α) (case 1, Table 4.2), with a comparison to the corresponding FLIGHTLAB model.

4.3.2 Roll Rate - Off-axis

In this section, the roll rate response is investigated, using the input and hyperparameter structure containing the current longitudinal pilot input with one lagged roll rate term utilising one hyperparameter (case 2, Table 4.2), as shown in equation (4.5), wherein this case y is the roll rate. The rotorcraft roll response to a longitudinal cyclic input is an off-axis response. Padfield stated that the off-axis motion is unpredictable [3]. It is, therefore, known to be

difficult to capture using existing physics based modelling techniques, as discussed by Haycock [61]. Attempts to improve the off-axis correlations have been undertaken. For example, Mansur and Tischler [62] used an empirical correction method to improve the off-axis correlation.

As with the pitch rate GP, simulated annealing was used to calculate the hyperparameters to govern the GP model. The accepted samples are shown in the form of a trace plot in Figure 4.17 and a histogram in Figure 4.18. For the initial OSAP and the FMP realisations without hyperparameter uncertainty, the mean of the accepted samples is used for the hyperparameters, which were, $\alpha = 0.972$ and $\beta = 31097$. To reach the stationary distribution (more detail in Section 2.1.3), 'burn-in' was conducted, which is shown as a trace plot in Figure 4.17.

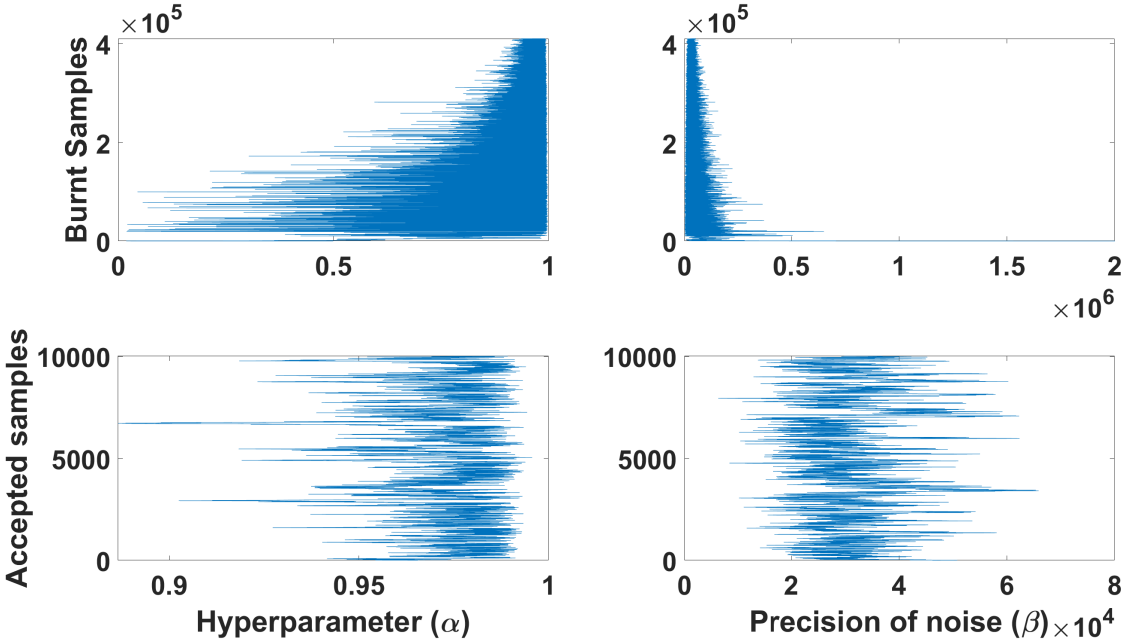


Figure 4.17: Simulated annealing burn-in and accepted samples used to create the roll rate GP model using the input structure containing the current longitudinal pilot input and one lagged roll rate term (case 2, Table 4.2).

Figure 4.19 displays the one step ahead predictions for the roll rate response, as previously discussed in Section 2.1.4 the one step ahead predictions are expected to be an accurate fit, as shown in the pitch rate response in Figure 4.13. The similarity measure between the one step ahead predictions and the flight test data is 0.06, where a value of less than one is deemed to be an excellent fit. As with the pitch rate OSAP, the confidence bounds are calculated to be three standard deviations from the mean and are just visible in Figure 4.19.

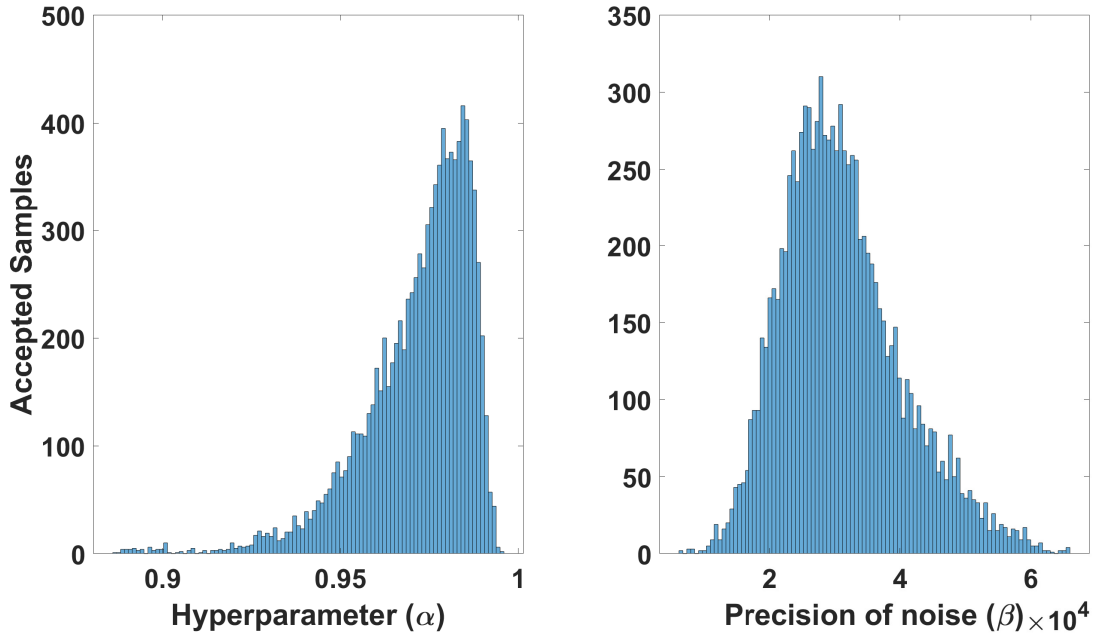


Figure 4.18: Simulated annealing results used to create the roll rate GP model using the input structure containing the current longitudinal pilot input and one lagged roll rate term (case 2, Table 4.2).

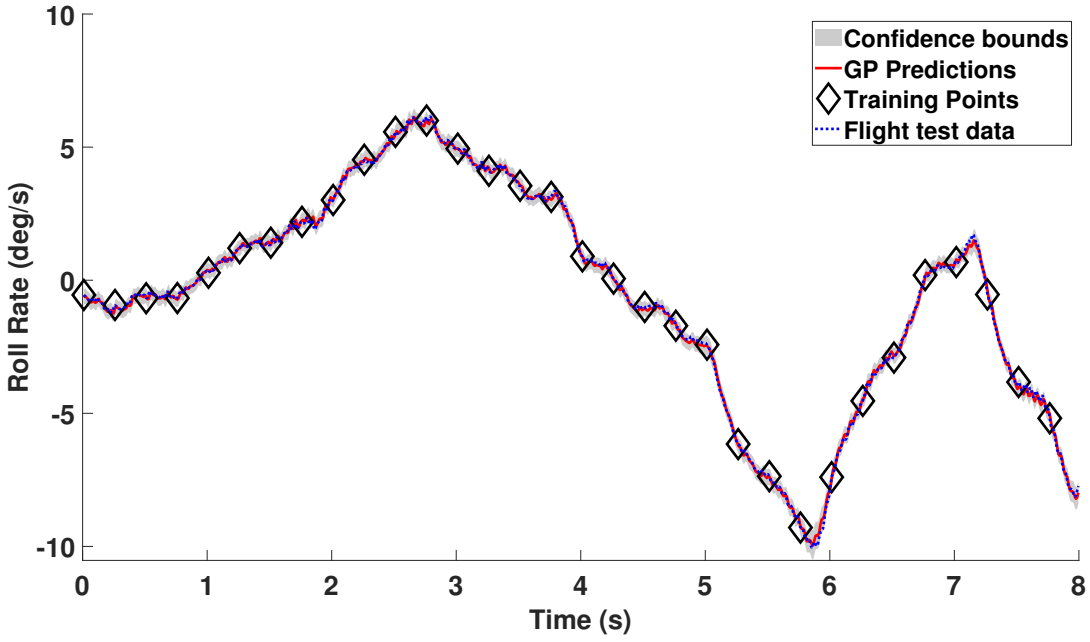


Figure 4.19: GP one step ahead roll rate predictions using the input structure containing the current longitudinal pilot input and one lagged roll rate term utilising one hyperparameter (α) (case 2, Table 4.2).

Figure 4.20 shows the predicted rotorcraft response of roll rate FMP realisations (red realisations) using the input structure containing the current longitudinal pilot input with one lagged roll rate term as the ARX input structure (case 2, Table 4.2) shown in equation (4.5). The main characteristics of the response were captured, particularly towards the end of the manoeuvre. The wider FMP realisations compared to the pitch rate response, Figure 4.14, are indicative of reduced confidence in the model predictions. The reduced confidence could be due to the unpredictability of the off-axis motion [3]. The predictions are for an off-axis response, which is difficult [3]. The GP captures the main characteristics of the response and therefore provides that the use of GPs to model rotorcraft dynamics is a feasible option.

The hyperparameter uncertainty is investigated in Figure 4.20 and displayed as FMP realisations (green realisations) on the same axes to that of the FMP realisations using the hyperparameters of $\alpha = 0.972$ and $\beta = 31097$ (red realisations). The overlapping realisations are displayed as grey. The hyperparameter uncertainty method uses the accepted samples shown in the histogram in Figure 4.18. As discussed in the previous section on the pitch rate response, the hyperparameters are randomly selected from the accepted samples for each realisation of the FMP. The Monte Carlo approach shows the effect of the hyperparameter uncertainty on the FMP realisations and can be easily compared to the FMP realisations, which used the hyperparameter calculated by the mean of the accepted samples. Visually, using Figure 4.20, the predictions made using hyperparameter uncertainty look very similar to the standard FMP realisations (using the mean of the accepted samples for the hyperparameters).

Figure 4.20 is a comparison between the FMP realisations of the roll rate predictions and the FLIGHTLAB model (black dashed line). The FLIGHTLAB model captures the initial part of the response very well but then becomes less accurate as the manoeuvre progresses. Conversely, the GP predictions do less well at the start of the manoeuvre but become increasingly accurate towards the final stages. The similarity measure of the GP FMP (without hyperparameter uncertainty) was 50.43, which outperforms the FLIGHTLAB model, having a similarity measure of 72.54. The values at first glance, seem poor; however, note that this is an off-axis response. The GP captures the essence of the manoeuvrer, and it outperforms the FLIGHTLAB model. It is, therefore, reasonable to investigate the use of GPs on rotorcraft dynamics further.

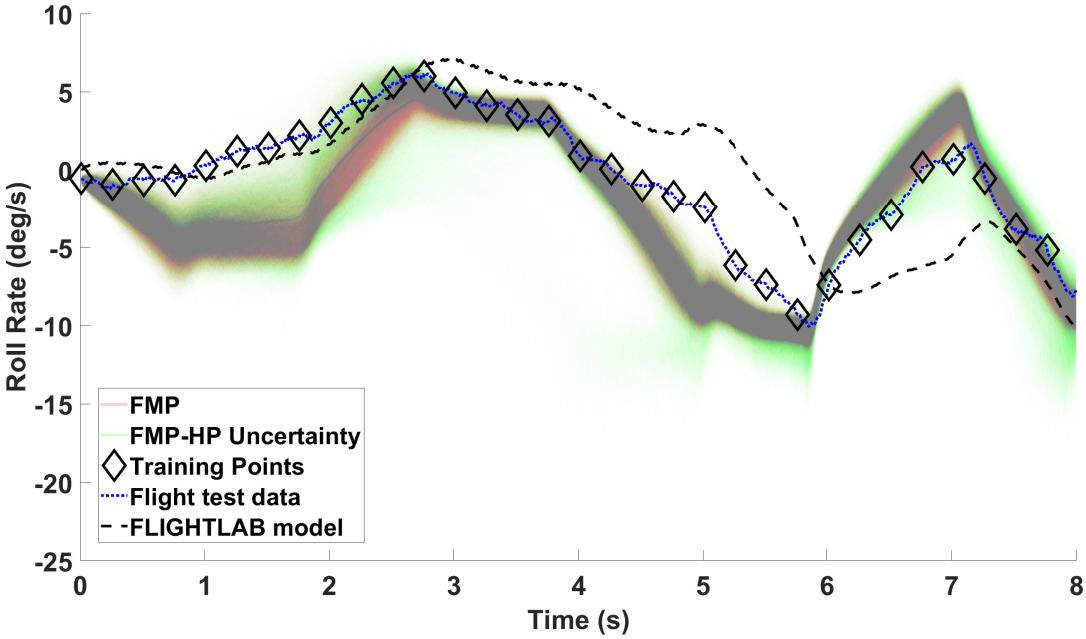


Figure 4.20: Realisations of the GP roll rate full model prediction using the input structure containing the current longitudinal pilot input and one lagged roll rate term utilising one hyperparameter (α) (case 1, Table 4.2), with a comparison to the corresponding FLIGHTLAB model.

4.4 Summary

In the current chapter, it was investigated whether machine learning regression can be used to improve the initial models for current rotorcraft flight simulators. It was found that the GP model outperformed the equivalent linear regression model. Different input structures for the linear regression model were investigated, and it was found that the ARX and ARMA produced the most accurate results. It was decided to use an ARX input structure for the GP models for the on-axis pitch rate, and off-axis roll rate and the effect of hyperparameter uncertainty on the GP FMP realisations was explored.

Table 4.3 enables ease of comparisons for the similarity measures for both of the cases investigated in this chapter. Note, all of the GP models outperform the FLIGHTLAB predictions in the cases shown, thus providing evidence that the GP models have the potential to improve the accuracy of the predictions for initial flight simulator models and that the ARX input structures are appropriate for the dataset, presenting motivation for investigating more complex ARX input structures in the subsequent chapters.

Manoeuvre	Prediction	Input structure	Case	Model	Similarity measure
3-2-1-1 Longitudinal input	Pitch rate	Longitudinal input with one lagged pitch rate term	1	OSA GP	**0.02**
				Mean FMP GP	**5.25**
				Mean FMP GP with hyperparameter uncertainty	**5.10**
				FLIGHTLAB	**75.02**
	Roll rate	Longitudinal input with one lagged roll rate term	2	OSA GP	**0.06**
				Mean FMP GP	**50.43**
				Mean FMP GP with hyperparameter uncertainty	**46.30**
				FLIGHTLAB	**72.54**

Table 4.3: Similarity measure for the GP models using the input structure containing current longitudinal input with relevant lagged output term utilising one hyperparameter. The GP models are shown in Chapter 4.

Using the results shown in Table 4.3, note that the FMP of on-axis response, pitch rate, produce more accurate results to that of the off-axis, roll rate. The result is expected, as the off-axis response is more unpredictable compared to that of the on-axis response [3]. However, GP FMP still produces better results than the FLIGHTLAB model. It is also important to note that the GP using an ARX input structure contains one current input of the longitudinal stick position; it is conceivable to assume that the pilot is using other inputs. It could, therefore, be beneficial to use more current pilot inputs in the ARX input structure for the GP, such as the lateral stick position, pedal position and collective lever. The use of more inputs is investigated in the next chapter.

It is noted in Section 4.3.1, that the FMP realisations do not encompass all of the flight test data. As discussed in Section 2.1.4, the FMP realisations should encompass all of the training data. There could be many reasons as to why the GP FMP realisations do not encompass all of the flight test data. One reason could be uncontrollable is external factors that affect the flight of the aircraft such as wind, which can not be captured by using the current inputs of the rotorcraft. As previously mentioned, the pilot potentially could be using other inputs to control the aircraft; this will affect how the GP predicts the response. This issue could be resolved using other ARX input structures, which is investigated more in the next chapter.

Using the evidence provided by the figures shown in Section 4.3, it is clear that incorporated hyperparameter uncertainty does not have a significant effect on the confidence bounds. The hyperparameter uncertainty also does not have a substantial effect on the average prediction, as shown by the similarity measure values in Table 4.3. Throughout, it was found that hyperparameter uncertainty was not significant. Therefore, hyperparameter uncertainty is not incorporated in the next chapters. It is also noted that the OSAP shown in the current section produces accurate predictions; this is expected due to the OSAP predicting one-time step ahead. For the sake of brevity, the OSAP for the GPs shown in the next chapters produce very accurate prediction and will not be displayed in the main text but can be seen in the Appendices.

Chapter 5

Application of Gaussian Process Regression to the Bo105 On-axis Responses

In the previous chapter, a feasibility study was conducted to see if a GP NARX model could accurately predict the on-axis response of pitch rate and off-axis response of the roll rate of a Bo105 rotorcraft. In the current chapter, the use of GPs to model rotorcraft dynamics is investigated further. Specifically, the use of different ARX input structures for the GP model (see Section 2.1.4 for more detail), referred to as GP input configurations, are investigated. The results shown in the current chapter are for the on-axis responses of the pitch rate, roll rate, yaw rate and heave, using the four manoeuvres discussed in Chapter 3. For each on-axis response, the different ARX input structures for the GP models are compared to each other as well as the corresponding FLIGHTLAB model and Bo105 flight test data. The current chapter investigates two scenarios for the input structure, firstly if the use of more pilot inputs for the ARX input structure improves the accuracy of the GP predictions compared to the use of only the primary input (for example, the longitudinal stick position is a primary input for the pitch rate). Secondly, if using more lagged outputs improves the accuracy of the predictions.

5.1 Gaussian Process Input Structures

In the current chapter, the four manoeuvres described in Chapter 3 are analysed using GPs with different input structures. For each manoeuvre, six different ARX input structures were investigated, each has a combination of one, two and three lagged output terms with different current pilot inputs, which include all of the current pilot inputs or the on-axis pilot input. For example, in the case of the pitch rate, the on-axis pilot input would be the longitudinal stick position. Each investigation is referred to as a 'case'. The ARX input structure using the on-axis plot input (cases 1,2 and 3, Tables 5.1, 5.3, 5.5 and 5.7) contains the primary input for the corresponding response with one, two or three lagged response terms. For example, to predict the nth pitch rate, the ARX input structure 'current longitudinal pilot input with one lagged pitch rate term' refers specifically to the input of the form:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \\ y_{n-1} \end{pmatrix} \tag{5.1}$$

where δ_n^x is the current longitudinal stick position and y_{n-1} is the lagged response (i.e. the pitch rate predicted at time $n-1$). The second ARX input structure contains all of the current pilot inputs with one, two and three lagged response terms (cases 4, 5 and 6, Tables 5.1, 5.3, 5.5 and 5.7). For example, to predict the nth pitch rate, the ARX input structure 'all of the current pilot inputs with one lagged pitch rate term' refers specifically to the input of the form:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \delta_n^y \\ \delta_n^p \\ \delta_n^o \\ \\ y_{n-1} \end{pmatrix} \tag{5.2}$$

where δ_n^y is the current lateral stick position, δ_n^p is the current pedal position and δ_n^o is the current collective lever.

In the current chapter, the comparison full model predictions (FMP) realisations figures are

the main focus of the analysis. As discussed in the previous chapter, the other figures, such as the simulated annealing results and the one step ahead predictions (OSAP) are shown in Appendix B.

It is expected that the predictions that will be the most accurate in terms of the similarity measure will contain two lagged response terms. The assumption that the most accurate model will contain two lagged response terms is due to the GP model capturing the underlying accelerations of the rotorcraft, which are associated with the second lagged response term. Two lagged terms are needed to emulate systems whose true response is governed by a second-order differential equation (those that include acceleration terms, in other words).

5.1.1 Pitch Rate

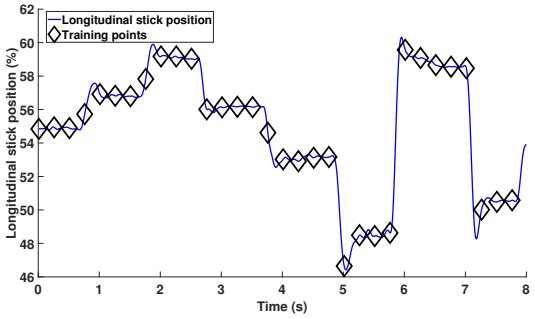
In the previous chapter, it was shown that a GP using an ARX input structure (GP NARX model) produced the most accurate results for the pitch rate response; however, the current input (longitudinal) and the number of lag terms (one lagged term) were fixed. In this section, the 3-2-1-1 longitudinal input manoeuvre is used to create GP models predicting pitch rate, using varied numbers of lag terms as inputs. The ARX input structure of the GP is investigated; each input structure is referred to as a case, shown in Table 5.1.

Manoeuvre	Prediction	Current inputs	Lagged outputs	Case
3-2-1-1 Longitudinal input	Pitch rate	Longitudinal input	One lagged pitch rate term	1
			Two lagged pitch rate terms	2
			Three lagged pitch rate terms	3
		Longitudinal, lateral, pedal and collective inputs	One lagged pitch rate term	4
			Two lagged pitch rate terms	5
			Three lagged pitch rate terms	6

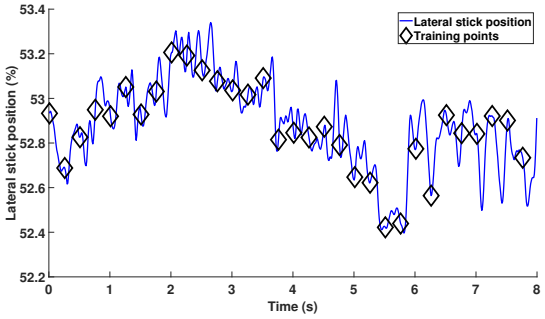
Table 5.1: GP configurations for the investigation of the pitch rate GP input structure

The 3-2-1-1 longitudinal input manoeuvre is shown in Figure 5.1a, displaying the excitation to the longitudinal input, which is the only pilot input used for cases 1, 2 and 3 in Table 5.1. Figure 5.1 displays the other pilot inputs used for the input structure 'all of the current pilot inputs' (case 4, 5 and 6, Table 5.1). The black diamonds indicate the location of the 32 training points that were used for model training. The 32 training points selected were chosen to be every 25th point in the dataset. The 32 training points chosen may not be the optimal

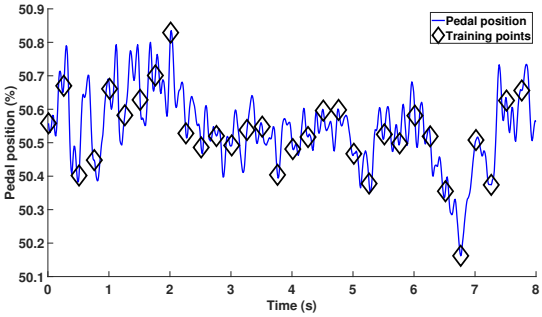
solution, but note that a method to find the optimum training points is discussed in more detail in Chapter 7. Reducing the number of data points from the full data set allows a subset of data points to be used for training. The full data set is used for predictions, and therefore allows interpolation between training points. Using the full data set for predictions enables validation on the model by using data that was not used in the training set.



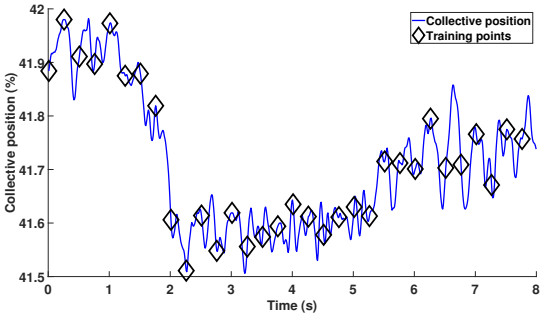
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 5.1: Bo105 rotorcraft inputs for the 3-2-1-1 longitudinal cyclic input with 32 training points.

The OSAP and simulated annealing figures are not shown in the main text; however, for the interested reader, they are displayed in the Appendix, B.1. The hyperparameters selected for the GPs in this section are chosen by using the mean of the accepted samples from the simulated annealing algorithm (more details can be found in Section 2.1.3). The hyperparameters for the cases were calculated to be:

- Case 1 - Current longitudinal pilot input with one lagged pitch rate term - $\alpha = 0.983$ and $\beta = 7.439 \times 10^4$
- Case 2 - Current longitudinal pilot input with two lagged pitch rate terms - $\alpha = 0.958$ and $\beta = 8.201 \times 10^5$

- Case 3 - Current longitudinal pilot input with three lagged pitch rate terms - $\alpha = 0.978$ and $\beta = 5.557 \times 10^5$
- Case 4 - All of the current pilot inputs with one lagged pitch rate term - $\alpha = 0.993$ and $\beta = 1.336 \times 10^5$
- Case 5 - All of the current pilot inputs with two lagged pitch rate terms - $\alpha = 0.992$ and $\beta = 5.946 \times 10^5$
- Case 6 - All of the current pilot inputs with three lagged pitch rate terms - $\alpha = 0.992$ and $\beta = 5.694 \times 10^5$

Figures 5.2, 5.3 and 5.4 display the realisations of pitch rate from the FMP when the GP used the current longitudinal pilot input with varying lagged pitch rate terms for the ARX input structure (case 1, 2 and 3, Table 5.1), shown by red realisations. For ease of comparison, Figures 5.2, 5.3 and 5.4 show pitch rate realisations from the FMP when the GP used all of the current pilot inputs (longitudinal, lateral, collective and pedal) with one, two and three lagged pitch rate terms for the ARX input structure (case 4, 5 and 6, Table 5.1), the realisations are given by a green semi-transparent line.

As shown in Section 2.1.4, a Monte Carlo analysis is required for FMP predictions. Figures 5.2, 5.3 and 5.4 show ensembles of predictions made by a Monte Carlo analysis that used 10,000 runs. The Monte Carlo results are plotted using semi-transparent lines, such that darker regions represent areas where more realisations were observed. The predictions made by the FLIGHTLAB model are shown as a dashed black line. The FMP GP realisations are not as accurate as the OSAP. However, this is to be expected given that, for this model, the previous prediction becomes an input to the next prediction and, hence, predictive uncertainty is carried through the simulation.

Comparing red (longitudinal input, case 1, Table 5.1) and green (all of the current pilot inputs, case 4, Table 5.1) realisations in Figure 5.2, it can be seen that there is a marginal improvement for the predictions in red realisations (longitudinal input, case 1, Table 5.1). However, the predictions are still quite similar. With regard to the physical law based FLIGHTLAB model,

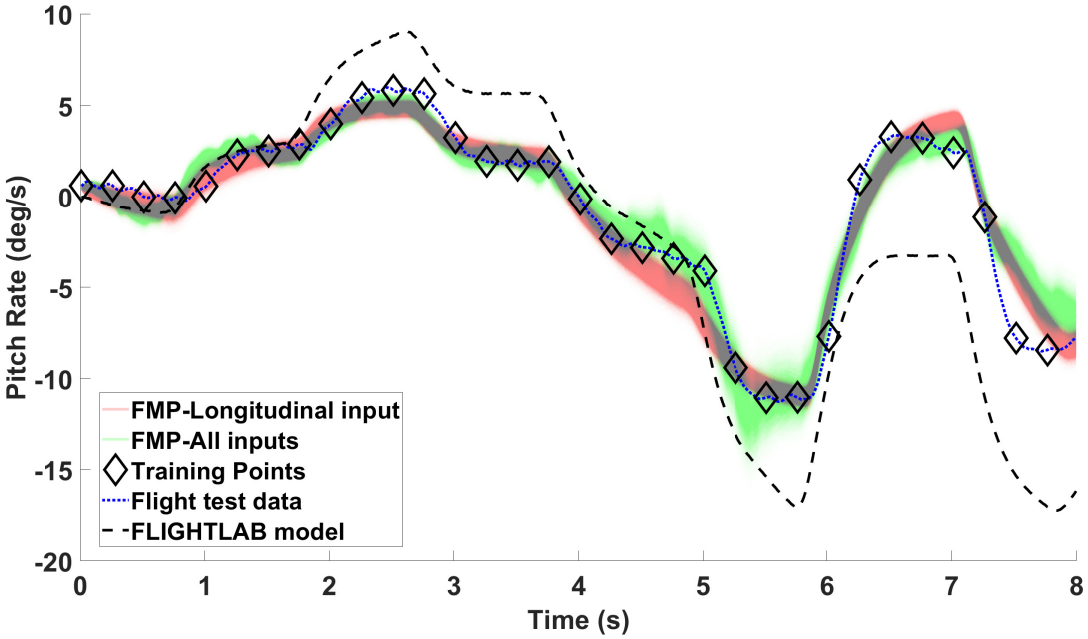


Figure 5.2: Realisations of the GP pitch rate full model prediction for the ARX input structure containing the current longitudinal pilot input with one lagged pitch rate term and all of the current pilot inputs with one lagged pitch rate term (cases 1 and 4 in Table 5.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

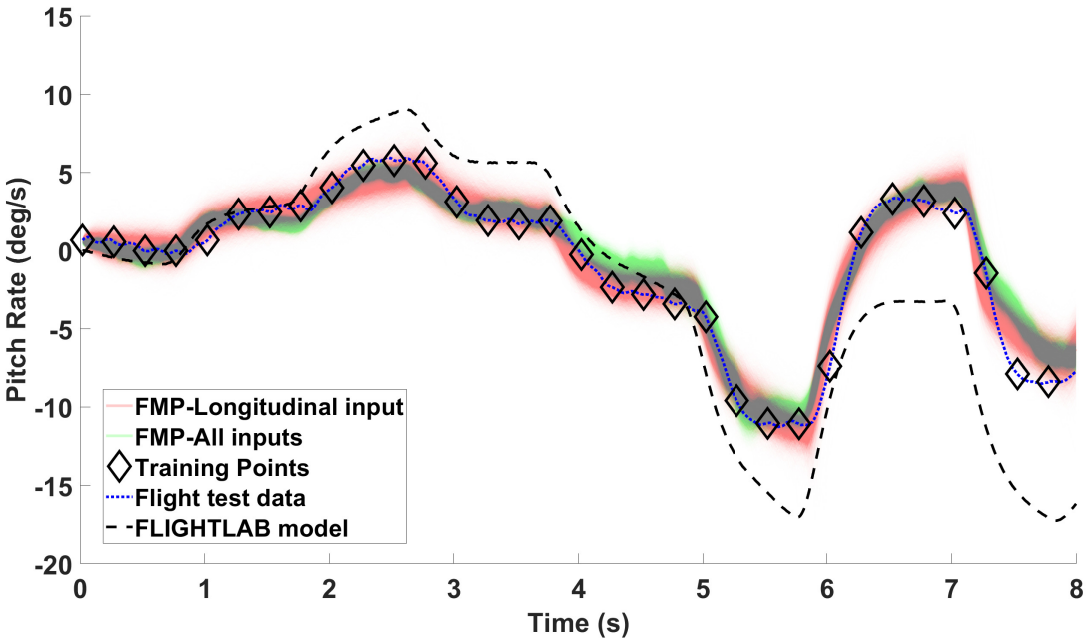


Figure 5.3: Realisations of the GP pitch rate full model prediction for the ARX input structure containing the current longitudinal pilot input with two lagged pitch rate terms and all of the current pilot inputs with two lagged pitch rate terms (cases 2 and 5 in Table 5.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

it can be seen that both models (GP and FLIGHTLAB) capture the essence of the real aircraft response. It is apparent that the GP model outperforms the FLIGHTLAB model in this case; in general, the GP predictions remain much closer to the flight test truth data throughout the manoeuvre, but particularly in its latter stages. Quantifying the improvement by using the average response of the FMP realisations in Figure 5.2. The similarity measure for the GP using all of the current pilot inputs with one lagged pitch rate term (green realisations) is 5.62 and for the GP utilising the input structure containing the current longitudinal pilot input with one lagged pitch rate term (red realisations) is 5.25. For the FLIGHTLAB model, the similarity measure is 75.02. These confirm what is easily observable, that, for the primary axis response, the GP model outperforms the FLIGHTLAB model. The GP using the input structure containing all of the current pilot inputs has a marginally worse similarity measure, this showing that the other current pilot inputs (lateral, pedal and collective) do not have a significant impact on the pitch rate.

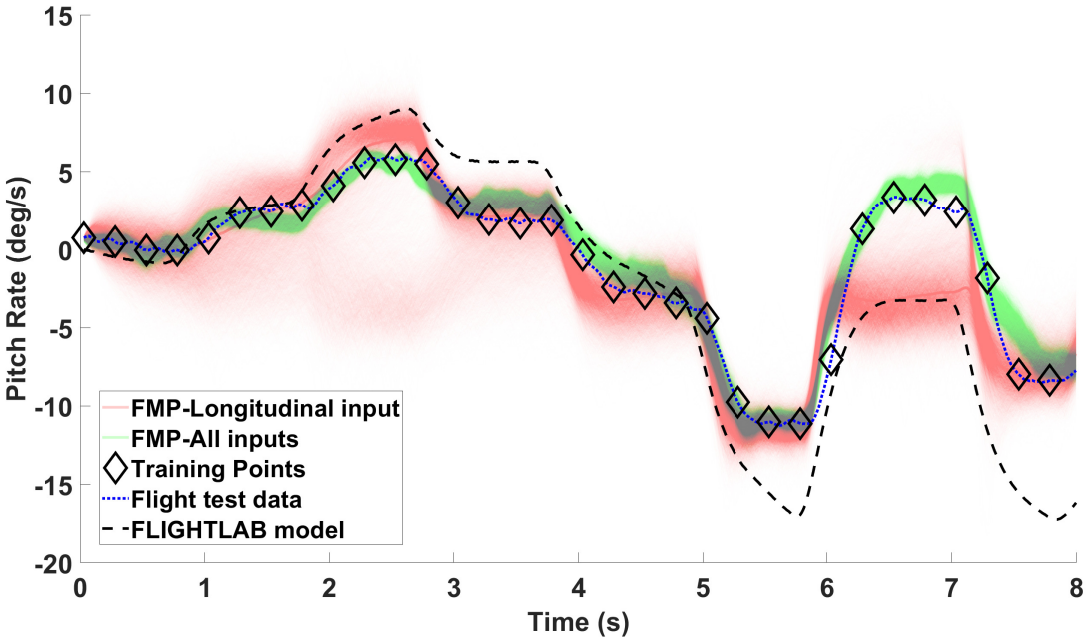


Figure 5.4: Realisations of the GP pitch rate full model prediction for the ARX input structure containing the current longitudinal pilot input with three lagged pitch rate terms and all of the current pilot inputs with three lagged pitch rate terms (cases 3 and 6 in Table 5.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

Figure 5.3 displays the FMP realisations for the input structures containing two lagged pitch rate terms. The green (all of the current pilot inputs, case 5, Table 5.1) and red (longitudinal

input, case 3, Table 5.1) realisations are very similar; however, it is difficult to judge which input structure produces the most accurate predictions. The input structure containing all of the current pilot inputs with two lagged pitch rate terms has a similarity measure of 5.52, and the similarity measure of the input structure containing the current longitudinal pilot input with two lagged pitch rate terms is 3.52. Comparing this to the FLIGHTLAB (similarity measure 75.02), both GP models outperform the FLIGHTLAB model predictions. The result provides evidence that the use of two lagged pitch rate terms improves the GP prediction, as already discussed, this is an expected result.

The FMP realisations from the GPs using the input structures containing the three lagged pitch rate terms is shown in Figure 5.4. The red realisations (longitudinal input, case 3, Table 5.1) are not as accurate as the green realisations (all of the current pilot inputs, case 6, Table 5.1). This result differs from the predictions in Figures 5.2 and 5.3. One hypothesis for the result is that the third lagged pitch rate term is causing the GP model with input structure containing the current longitudinal pilot input to be overfitting, which would make the model very sensitive to data not seen before, i.e. full model predictions. It seems that the GP model is identifying a relationship that holds for the OSAP but does not generalise beyond that, which would be the case when applying full model predictions. The similarity measure for the two GP models is 20.52 for the input structure containing the current longitudinal pilot input with three lagged pitch rate terms, and 4.07 for the input structure of all of the current pilot inputs with three lagged pitch rate terms. As with the previous figures, the GP outperforms the FLIGHTLAB model.

In Figures 5.2, 5.3 and 5.4, it can be seen that the ensemble of GP predictions does not always encompass all of the flight test data. Reasons for this discrepancy are discussed further in Chapter 10.

Crucially, all the GP FMP outperform the physical law based FLIGHTLAB predictions; the GP appears better able to capture the dynamics of the flight test data. Table 5.2 displays the similarity measure for pitch rate cases, which shows that the GP model with the input structure containing the current longitudinal pilot input with two lagged pitch rate terms created the most accurate prediction. One reason for the result could be that the other pilot inputs do

Prediction	Current inputs	Lagged outputs	Case	Model	Similarity measure
Pitch rate	Longitudinal input	One lagged pitch rate term	1	OSA GP	0.02
				Mean FMP GP	5.25
		Two lagged pitch rate terms	2	OSA GP	0.01
				Mean FMP GP	**3.52**
		Three lagged pitch rate terms	3	OSA GP	0.01
				Mean FMP GP	20.52
	Longitudinal, Lateral, Pedal and Collective inputs	One lagged pitch rate term	4	OSA GP	0.02
				Mean FMP GP	5.62
		Two lagged pitch rate terms	5	OSA GP	0.03
				Mean FMP GP	5.52
		Three lagged pitch rate terms	6	OSA GP	0.02
				Mean FMP GP	4.07
				FLIGHTLAB	75.02

Table 5.2: The similarity measures of the different ARX input structures with varying pilot current inputs and lagged pitch rate terms for the GP model predicting pitch rate.

not have as much of an effect on the pitch rate as the longitudinal stick position, which seems feasible as the pitch rate is the on-axis response of the longitudinal stick position. Observing Figure 5.1, it can be seen that the other pilot inputs deviate a maximum of one percent. This small change in the other pilot inputs may not have had a significant effect on the pitch rate. Including the other current pilot inputs may be adding inputs that are mostly 'noise', and could be causing the GP to overfit.

5.1.2 Roll Rate

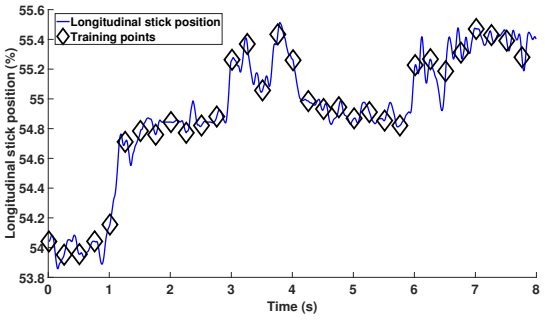
In this section, the 3-2-1-1 lateral input manoeuvre is used to create GP models for predicting the roll rate. The ARX input structure of the GP is investigated; the different input cases are shown in Table 5.3.

Figure 5.5 displays the inputs used for the GPs created for the roll rate response. The ARX input structure containing all of the current pilot inputs (cases 4, 5 and 6, Table 5.3), uses inputs shown in all four subplots given in Figure 5.5. The ARX input structure that is denoted

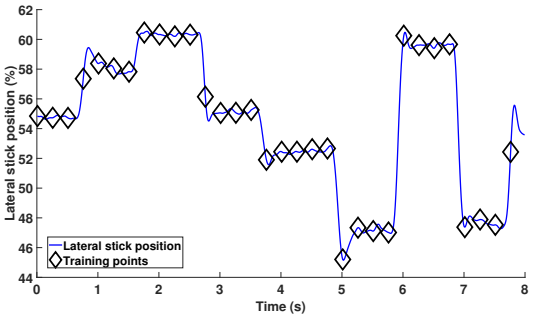
Manoeuvre	Prediction	Current inputs	Lagged outputs	Case
3-2-1-1 Lateral input	Roll rate	Lateral input	One lagged roll rate term	1
			Two lagged roll rate terms	2
			Three lagged roll rate terms	3
		Longitudinal, lateral, pedal and collective inputs	One lagged roll rate term	4
			Two lagged roll rate terms	5
			Three lagged roll rate terms	6

Table 5.3: GP configurations for the investigation of the roll rate GP input structure

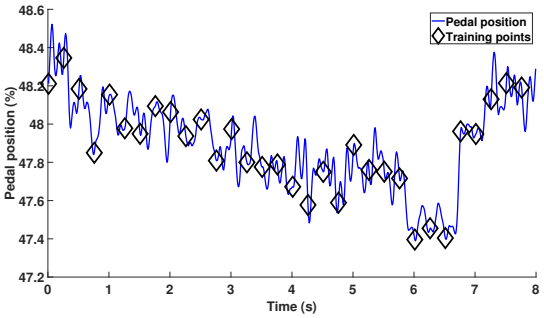
as the lateral input (case 1, 2 and 3, Table 5.3) only uses the lateral stick position shown in Figure 5.5b as the pilot input for the ARX input structure. The roll rate prediction, which uses the 3-2-1-1 lateral input manoeuvre shown in Figure 5.5b, where the diamonds are the 32 training points for the GP. As previously discussed the chosen training points shown in Figure 5.5 may not be the optimal solution. The optimal choice of training points is not discussed until Chapter 7.



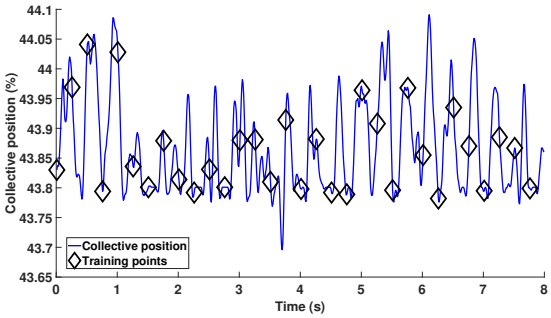
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 5.5: Bo105 rotorcraft inputs for the 3-2-1-1 lateral cyclic input with 32 training points.

The OSAP and simulated annealing results for the GPs in the current section are shown in the Appendix B.2. The hyperparameters for the six GPs were calculated (using the mean of the

accepted samples) to be:

- Case 1 - Current lateral pilot input with one lagged roll rate term - $\alpha = 0.935$ and $\beta = 2.341 \times 10^4$
- Case 2 - Current lateral pilot input with two lagged roll rate terms - $\alpha = 0.968$ and $\beta = 3.586 \times 10^4$
- Case 3 - Current lateral pilot input with three lagged roll rate terms - $\alpha = 0.961$ and $\beta = 3.980 \times 10^4$
- Case 4 - All of the current pilot inputs with one lagged roll rate term - $\alpha = 0.992$ and $\beta = 1.777 \times 10^4$
- Case 5 - All of the current pilot inputs with two lagged roll rate terms - $\alpha = 0.992$ and $\beta = 4.602 \times 10^4$
- Case 6 - All of the current pilot inputs with three lagged roll rate terms - $\alpha = 0.995$ and $\beta = 3.996 \times 10^4$

Figures 5.7, 5.8 and 5.9 show the FMP realisations for the ARX input structure that contained the current lateral pilot input with one, two and three lagged roll rate terms (cases 1, 2 and 3, Table 5.3), shown by the red realisations. It can be seen that in Figure 5.7 that the red realisations deviate from the flight test data by roughly 20 deg/s, particularity in the middle stages (3 to 6 seconds). This type of result can arise when an incomplete input structure is utilised as, in this relatively low-dimensional input space, the 'closeness' of new inputs to the training data may be erroneously estimated relative to if a higher dimensionality input space has been used. To illustrate the point, consider a 2D example where it is shown that using an incomplete input structure can lead to an erroneous estimation of the 'closeness' between the points. Figure 5.6 illustrates an example where an incomplete input structure is utilised (only knowing $x_1^{(1)}$ and $x_1^{(2)}$) the distance between the points is smaller than the complete input structure which utilities the data points of $x_1^{(1)}$, $x_1^{(2)}$, $x_2^{(1)}$ and $x_2^{(2)}$.

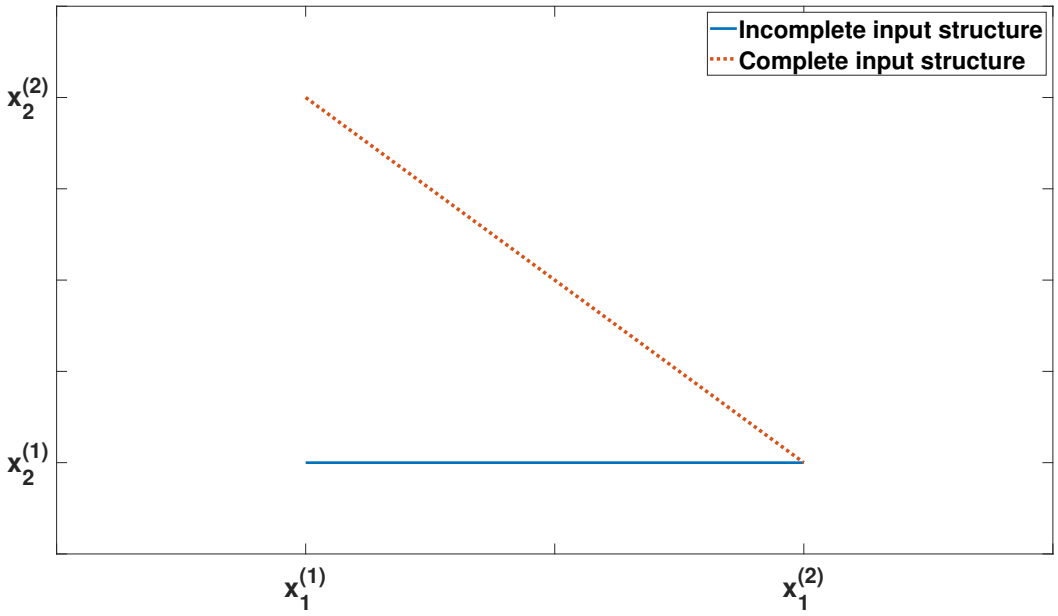


Figure 5.6: Comparing the distances between points when a complete and incomplete input structure is utilised.

The green realisations in Figures 5.7, 5.8 and 5.9 show the FMP realisations for the ARX input structure that utilised all of the current pilot inputs with one, two and three lagged roll rate terms (case 4, 5 and 6, Table 5.3). Firstly, note the improvement compared to the red realisations. The green realisations in Figures 5.7, 5.8 and 5.9 are significantly better than the red realisations. The improvement provides evidence that, in the case of the roll rate response, all of the current pilot inputs are required to capture the roll rate dynamics. The result provides evidence that there is cross-coupling between other pilot inputs and the roll rate response. It can be seen in Figure 5.5 that the maximum deflection of the other inputs is roughly 1.5 percent. The largest was in the longitudinal stick position, as mentioned previously, one of the largest cross-coupling is between the roll and pitch axis [3]. Padfield [3] states that in hover, the off-axis roll response from a pitch input can be as large as the on-axis response, confirming what the GP models indicate that the other pilot inputs affect the response of roll rate. Therefore, the use of just the on-axis lateral stick position is not enough to capture the true response during the manoeuvre.

In Figures 5.7, 5.8 and 5.9, the FLIGHTLAB model is compared to the GP FMP realisations and the flight test data. The GP model outperforms the FLIGHTLAB model. The similarity

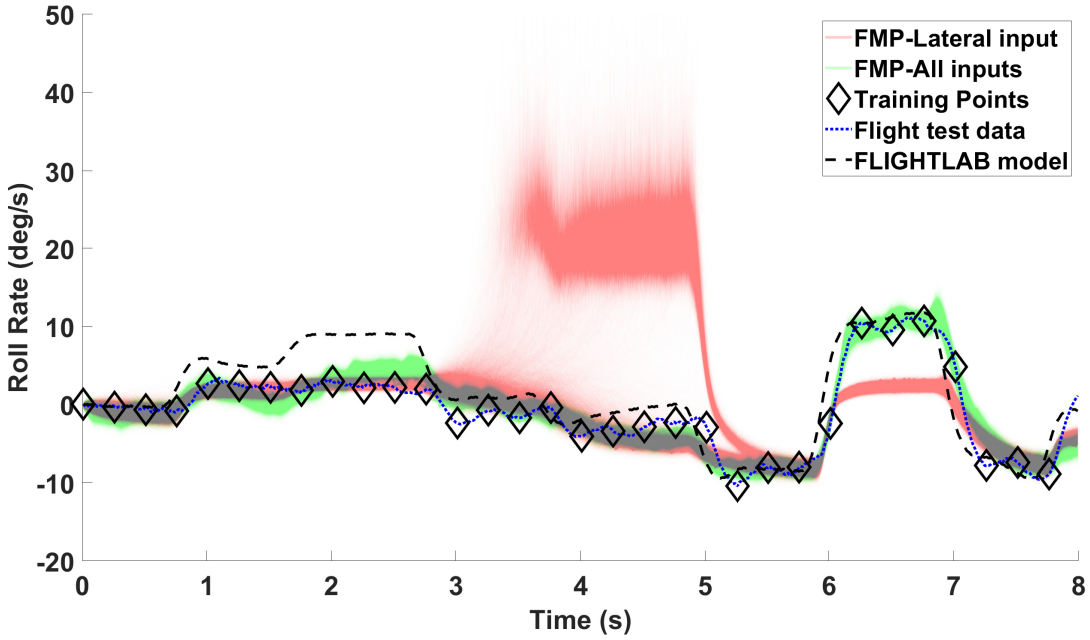


Figure 5.7: Realisations of the GP roll rate full model prediction for the ARX input structure containing the current lateral pilot input with one lagged roll rate term and all of the current pilot inputs with one lagged roll rate term (cases 1 and 4 in Table 5.3) with comparison to the FLIGHTLAB model and corresponding flight test data.

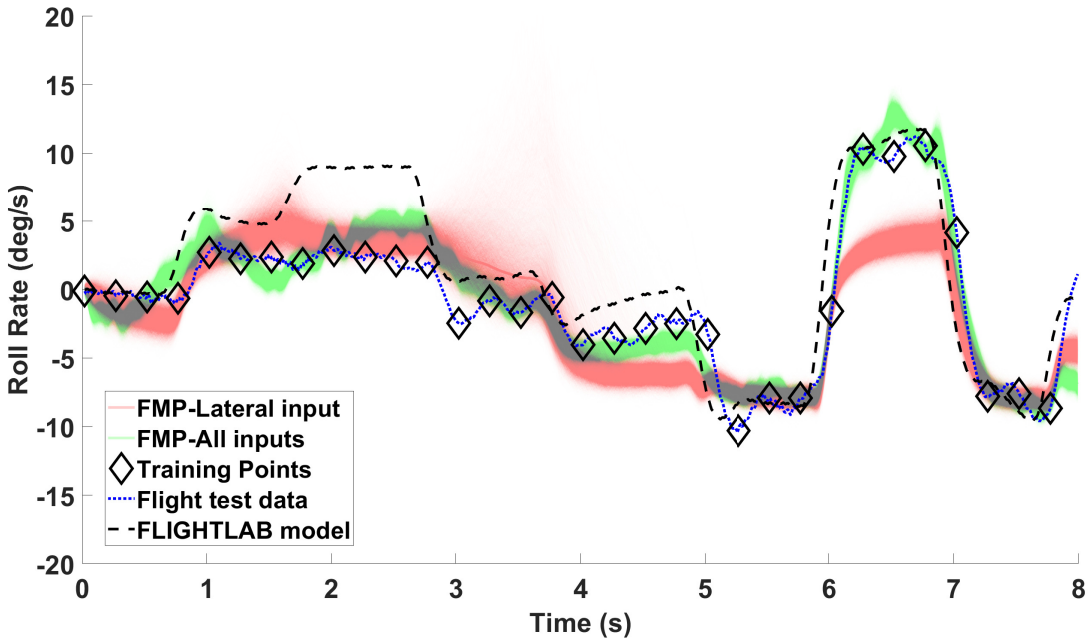


Figure 5.8: Realisations of the GP roll rate full model prediction for the ARX input structure containing the current lateral pilot input with two lagged roll rate terms and all of the current pilot inputs with two lagged roll rate terms (cases 2 and 5 in Table 5.3) with comparison to the FLIGHTLAB model and corresponding flight test data.

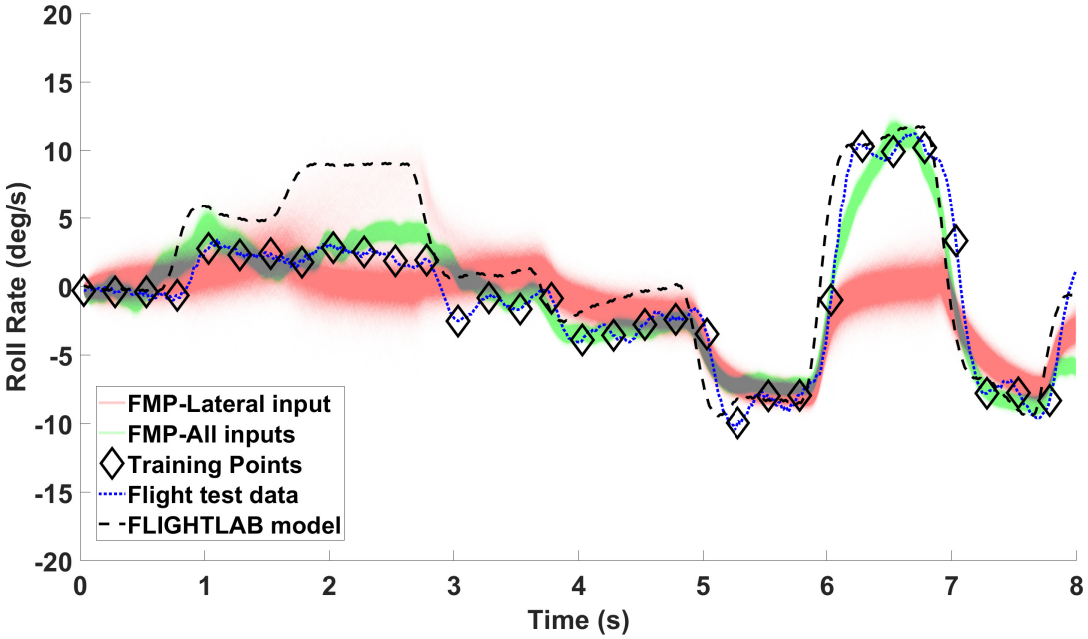


Figure 5.9: Realisations of the GP roll rate full model prediction for the ARX input structure containing the current lateral pilot input with three lagged roll rate terms and all of the current pilot inputs with three lagged roll rate terms (cases 3 and 6 in Table 5.3) with comparison to the FLIGHTLAB model and corresponding flight test data.

measure of the mean of the FMP realisations for each of the cases is shown in Table 5.4. It is evident from using Figures 5.7, 5.8, 5.9 and Table 5.4 that the input structures containing all of the current pilot inputs create the most accurate GP predictions. However, the number of lagged roll rate terms did not notably affect the predictions. It may be due to that using all of the current pilot inputs is more critical than the number of lagged terms. The GPs utilising the input structure of all of the current pilot inputs outperform the FLIGHTLAB predictions, supporting the hypothesis that a GP model can create more accurate predictions than an initial FLIGHTLAB model.

5.1.3 Yaw Rate

The ARX input structure for the GP predicting the yaw rate using the 3-2-1-1 pedal input manoeuvre is investigated in the current section. Each ARX input structure is referred to as a case; the 6 cases are displayed in Table 5.5.

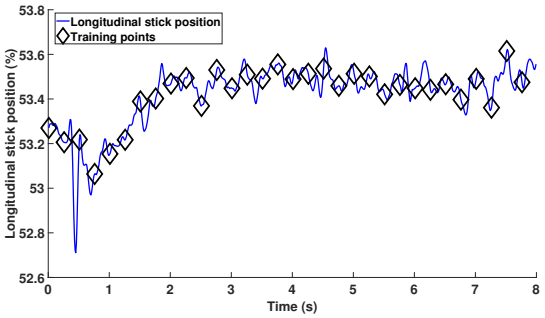
Figure 5.10 displays the inputs used for the ARX input structure for the on-axis yaw rate response. The ARX input structure containing all of the current pilot inputs (case 4,5 and 6,

Prediction	Current inputs	Lagged outputs	Case	Model	Similarity measure
Roll rate	Lateral input	One lagged roll rate term	1	OSA GP	0.15
				Mean FMP GP	58.65
		Two lagged roll rate terms	2	OSA GP	0.09
				Mean FMP GP	32.82
		Three lagged roll rate terms	3	OSA GP	0.11
				Mean FMP GP	48.80
	Longitudinal, Lateral, Pedal and Collective inputs	One lagged roll rate term	4	OSA GP	0.13
				Mean FMP GP	**10.30**
		Two lagged roll rate terms	5	OSA GP	0.15
				Mean FMP GP	**10.83**
		Three lagged roll rate terms	6	OSA GP	0.24
				Mean FMP GP	**10.47**
				FLIGHTLAB	42.43

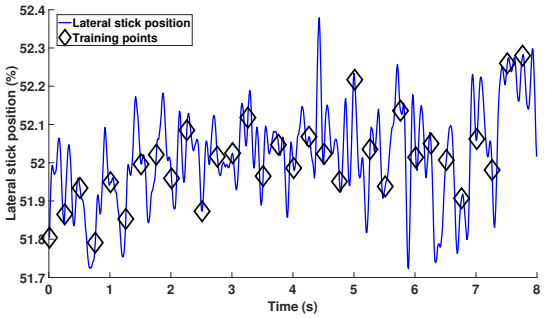
Table 5.4: The similarity measures of the different ARX input structures with varying pilot current inputs and lagged roll rate terms for the GP model predicting roll rate.

Manoeuvre	Prediction	Current inputs	Lagged outputs	Case
3-2-1-1 Pedal input	Yaw rate	Pedal input	One lagged yaw rate term	1
			Two lagged yaw rate terms	2
			Three lagged yaw rate terms	3
		Longitudinal, lateral, pedal and collective inputs	One lagged yaw rate term	4
			Two yaw roll rate terms	5
			Three yaw roll rate terms	6

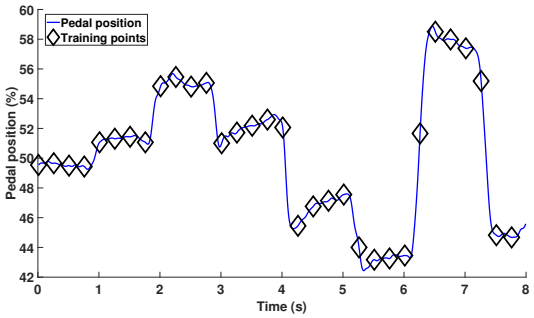
Table 5.5: GP configurations for the investigation of the yaw rate GP input structure



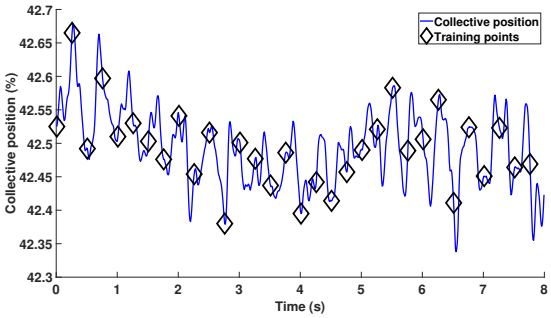
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 5.10: Bo105 rotorcraft inputs for the 3-2-1-1 pedal input with 32 training points.

Table 5.5) uses all of the inputs in Figure 5.10. The ARX input structure that consists of the current pedal pilot input (case 1, 2 and 3, Table 5.5) use the 3-2-1-1 pedal input manoeuvre in Figure 5.10c.

As with the previous section, for the interested reader, the OSAP and simulated annealing results are shown in the Appendix B.3. The hyperparameters calculated for the GPs in the section using the mean of the simulated annealing accepted samples are:

- Case 1 - Current pedal pilot input with one lagged yaw rate term - $\alpha = 0.952$ and $\beta = 6.925 \times 10^4$
- Case 2 - Current pedal pilot input with two lagged yaw rate terms - $\alpha = 0.948$ and $\beta = 7.153 \times 10^4$
- Case 3 - Current pedal pilot input with three lagged yaw rate terms - $\alpha = 0.949$ and $\beta = 7.297 \times 10^4$
- Case 4 - All of the current pilot inputs with one lagged yaw rate - $\alpha = 0.991$ and $\beta = 6.836 \times 10^4$

- Case 5 - All of the current pilot input with two lagged yaw rate terms - $\alpha = 0.991$ and $\beta = 7.222 \times 10^4$
- Case 6 - All of the current pilot inputs with three lagged yaw rate terms - $\alpha = 0.991$ and $\beta = 7.252 \times 10^4$

Figures 5.11, 5.12 and 5.13 display the GP full model realisations for the yaw rate prediction using 3-2-1-1 pedal input (case 1, 2 and 3, Table 5.5), which utilised the ARX input structure containing the current pedal pilot input with one, two and three lagged yaw rate terms, given as red realisations. The GPs struggle to accurately predict the yaw rate using only the current pedal pilot input. This result provides evidence that all of the current pilot inputs are required to capture the dynamics of the system; this was also found in the previous section for the roll rate predictions.

Figures 5.11, 5.12 and 5.13 display the FMP realisations for the yaw rate prediction that used the ARX input structure containing all of the current pilot inputs with one, two and three lagged yaw rate terms (case 4, 5 and 6, Table 5.5), denoted by the green realisations. The GP FMP realisations are an acceptable fit and do capture the flight dynamics. The yaw rate predictions are not as accurate as the pitch and roll rate prediction. However, it is also noticeable that the FLIGHTLAB model performance is poor, especially in the latter part of the prediction.

Similar to the roll rate response, the yaw rate response requires the use of all of the current pilot inputs to capture the response accurately; this is clear in Figures 5.11, 5.12 and 5.13. In Figure 5.11 from time 6 seconds onwards, the red realisations (pedal input) struggle to capture the dynamics and are opposite to that of the flight test data. In Figures 5.12 and 5.13, the GP red (pedal input) realisations fail to capture the dynamics in the region of 4 to 6 seconds, which would suggest that in the latter stages of the response (4 seconds onwards), the pedal position is not providing enough information to provide accurate predictions. It is hypothesised that other pilot inputs must be affecting the yaw rate response. Figure 5.10 displays all of the current pilot inputs used for the GP model using an ARX input structure. Apart from the pedal position deviation from time 6 seconds onwards, the other pilot inputs do not show a significant deviation. However, the combined pilot input deviations may have

affected the yaw rate response, as the GP models using the ARX input structure that contains all of the current pilot inputs capture the dynamics during the latter stages of the yaw rate response. As discussed in the previous section, a GP model using an incomplete input structure may inaccurately estimate the 'closeness' between new inputs and the training data. The red realisations that use input structure containing the current pedal pilot input with two lagged yaw rate terms, Figure 5.12, and with three lagged yaw rate terms, Figure 5.13 are very similar. However, when one lagged yaw rate term is used the red realisation differ, this is displayed in Figure 5.11, thus indicating that more than one lagged term is required to capture the dynamics accurately. However, the red realisations in Figures 5.12 and 5.13 do still miss some of the dynamics, especially between time 4 to 6 seconds. The missed dynamics is better captured when the GP utilities the input structure containing all of the current pilot inputs.

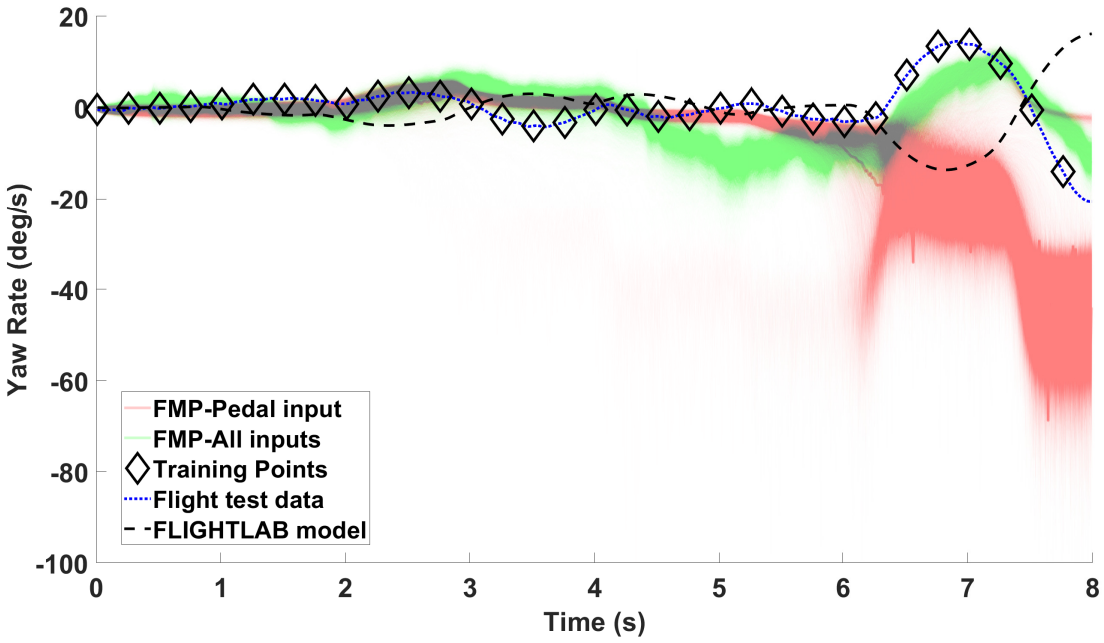


Figure 5.11: Realisations of the GP yaw rate full model prediction for the ARX input structure containing the current pedal pilot input with one lagged yaw rate term and all of the current pilot inputs with one lagged yaw rate term (cases 1 and 4 in Table 5.5) with comparison to the FLIGHTLAB model and corresponding flight test data.

The similarity measure for all the models predicting the yaw rate in Figures 5.11, 5.12 and 5.13 is shown in Table 5.6. Using the similarity measure table, it is clear that the GP using the input structure containing all of the current pilot inputs creates the most accurate model. The best model in terms of similarity measure between the mean FMP realisations and the flight test

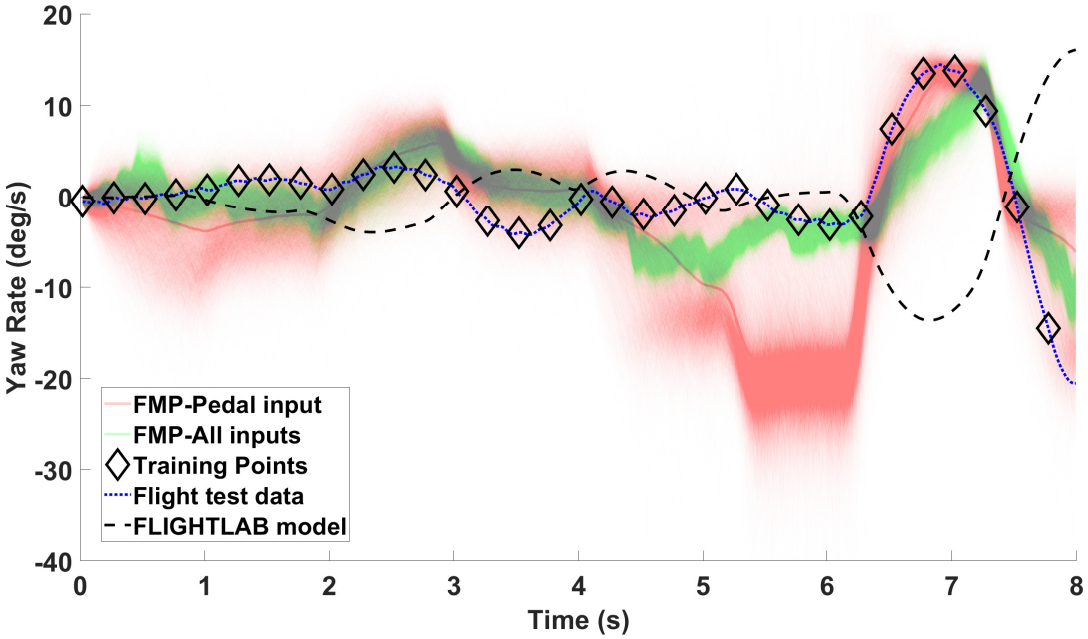


Figure 5.12: Realisations of the GP yaw rate full model prediction for the ARX input structure containing the current pedal pilot input with two lagged yaw rate terms and all of the current pilot inputs with two lagged yaw rate terms (cases 2 and 5 in Table 5.5) with comparison to the FLIGHTLAB model and corresponding flight test data.

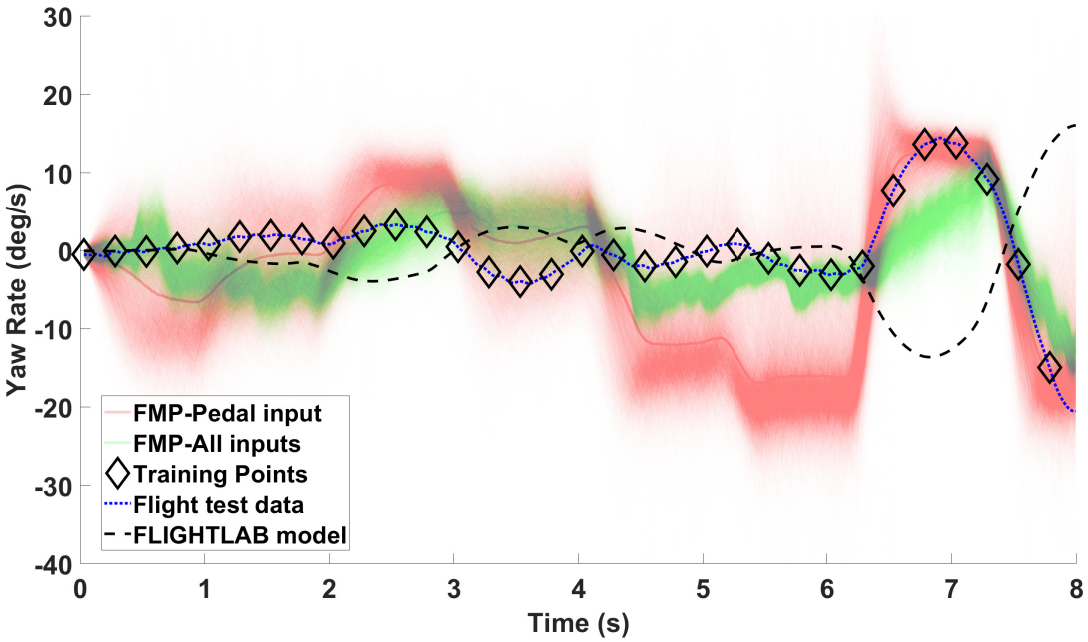


Figure 5.13: Realisations of the GP yaw rate full model prediction for the ARX input structure containing the current pedal pilot input with three lagged yaw rate terms and all of the current pilot inputs with three lagged yaw rate terms (cases 3 and 6 in Table 5.5) with comparison to the FLIGHTLAB model and corresponding flight test data.

data is the GP utilising the input structure containing all of the current pilot inputs with two lagged output terms. The similarity measure is 53.26 (compared to 370.08 for FLIGHTLAB). The similarity measures support the claim that the GP can create a more accurate initial model for the yaw rate response.

Prediction	**Current inputs**	**Lagged outputs**	**Case**	**Model**	**Similarity measure**
Yaw rate	Pedal input	One lagged yaw rate term	1	OSA GP	0.09
				Mean FMP GP	468.28
		Two lagged yaw rate terms	2	OSA GP	0.01
				Mean FMP GP	122.84
		Three lagged yaw rate terms	3	OSA GP	0.01
				Mean FMP GP	119.54
	Longitudinal, Lateral, Pedal and Collective inputs	One lagged yaw rate term	4	OSA GP	0.05
				Mean FMP GP	104.12
		Two lagged yaw rate terms	5	OSA GP	0.08
				Mean FMP GP	**53.26**
		Three lagged yaw rate terms	6	OSA GP	0.06
				Mean FMP GP	69.04
				FLIGHTLAB	370.08

Table 5.6: The similarity measures of the different ARX input structures with varying pilot current inputs and lagged yaw rate terms for the GP model predicting yaw rate.

5.1.4 Heave

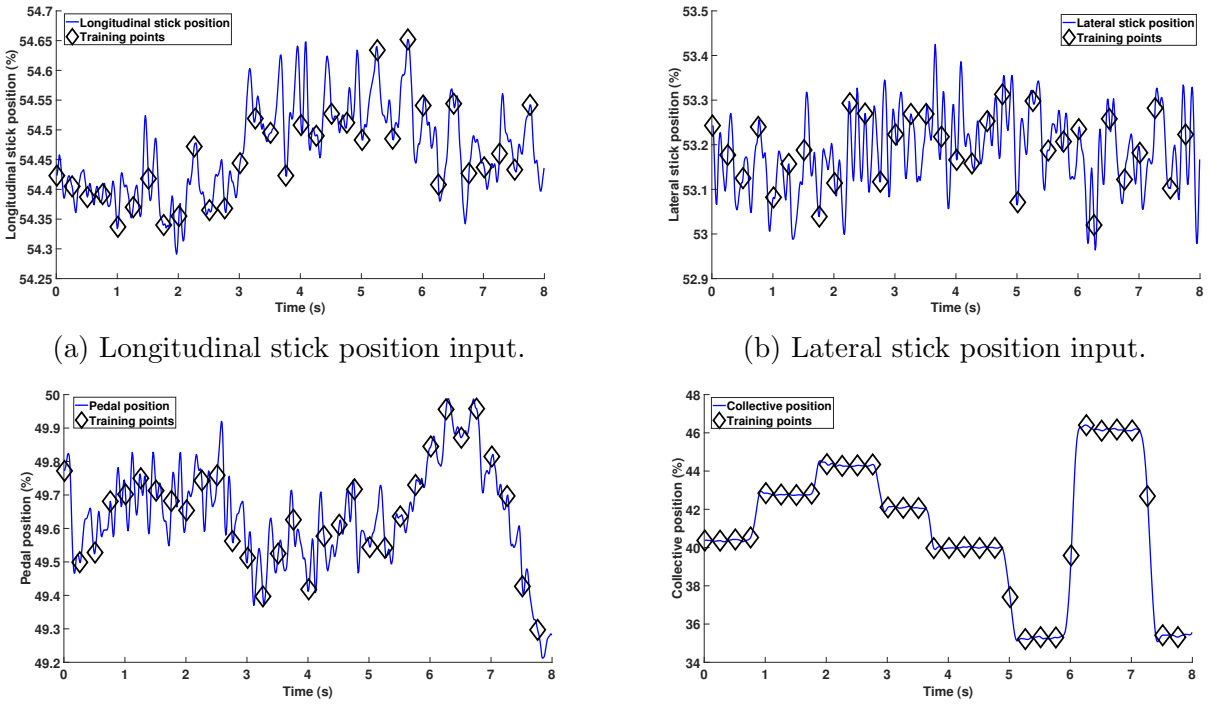
In this section, the 3-2-1-1 collective input manoeuvre is used to create GP models predicting heave. The ARX input structure of the GP is investigated; the different input structures are shown in Table 5.7.

Figure 5.14d displays the 3-2-1-1 collective input manoeuvre, which is used for cases 1, 2 and 3 in Table 5.7 (shown in this section). All of the current pilot inputs are displayed in Figure 5.14, which are used in cases 4,5 and 6 in Table 5.7. The 32 training points for the heave GPs as shown as black diamonds.

The OSAP and simulated annealing results are shown in the Appendix B.4. The hyperparame-

Manoeuvre	Prediction	Current inputs	Lagged outputs	Case
3-2-1-1 Collective input	Heave	Collective input	One lagged heave term	1
			Two lagged heave terms	2
			Three lagged heave terms	3
		Longitudinal, lateral, pedal and collective inputs	One lagged heave term	4
			Two yaw heave terms	5
			Three yaw heave terms	6

Table 5.7: GP configurations for the investigation of the yaw rate GP input structure



(a) Longitudinal stick position input.

(b) Lateral stick position input.

(c) Pedal position input.

(d) Collective position input.

Figure 5.14: Bo105 rotorcraft inputs for the 3-2-1-1 collective input with 32 training points.

ters for the GPs in this section were calculated by the mean of the simulated annealing results, and were:

- Case 1 - Current collective lever pilot input with one lagged heave term - $\alpha = 0.964$ and $\beta = 1.313 \times 10^4$
- Case 2 - Current collective lever pilot with two lagged heave terms - $\alpha = 0.974$ and $\beta = 2.342 \times 10^3$
- Case 3 - Current collective lever pilot with three lagged heave terms - $\alpha = 0.991$ and $\beta = 1.784 \times 10^3$
- Case 4 - All of the current pilot inputs with one lagged heave term - $\alpha = 0.986$ and $\beta = 1.192 \times 10^4$
- Case 5 - All of the current pilot inputs with two lagged heave terms - $\alpha = 0.986$ and $\beta = 2.620 \times 10^3$
- Case 6 - All of the current pilot inputs with three lagged heave terms - $\alpha = 0.983$ and $\beta = 1.868 \times 10^3$

The FMP realisations of the GP that utilised the ARX input structure containing the current collective lever pilot input with one, two and three lagged heave terms (case 1, 2 and 3, Table 5.7) are shown in Figures 5.15, 5.16 and 5.17 as red realisations. It is quite clear visually that the GP struggles to capture the dynamics. However, it is also apparent that the FLIGHTLAB model is far from the flight test data.

The GP FMP realisations that utilise the input structure containing all of the current pilot inputs with one, two and three lagged heave terms (cases 4, 5 and 6, Table 5.7) are denoted by green realisations in Figures 5.15, 5.16 and 5.17. It is evident that there is an improvement when all of the current pilot inputs are utilised in the FMP realisations. The analysis of Figures 5.15 and 5.17, where the GP input structures utilise one and three lagged heave terms, respectively, supports the statement that all of the current pilot inputs are required to capture the dynamics of the system accurately. However, when the GP input structure contains two lagged heave

terms, Figure 5.16, the result does not support the statement that all of the current pilot inputs are required to capture the dynamics accurately as the green realisations do not capture the dynamics of the heave response.

The FMP realisations for the GP with the ARX input structure containing the collective lever pilot input with one lagged heave term provide wildly inaccurate realisations and seem to produce divergent predictions. As previously discussed, when using an incomplete input structure is utilised the 'closeness' of new inputs to the training data may be inaccurately estimated. Another reason may be due to the uncertainty that is propagated across the time series in the FMP, as the predicted output becomes a GP input. Another possible reason is that the other pilot inputs could have also contributed to the heave response. If the other pilot inputs (such as longitudinal, lateral or pedal) affected the heave response, it is, therefore, a plausible outcome to require the other pilot inputs to model the response accurately.

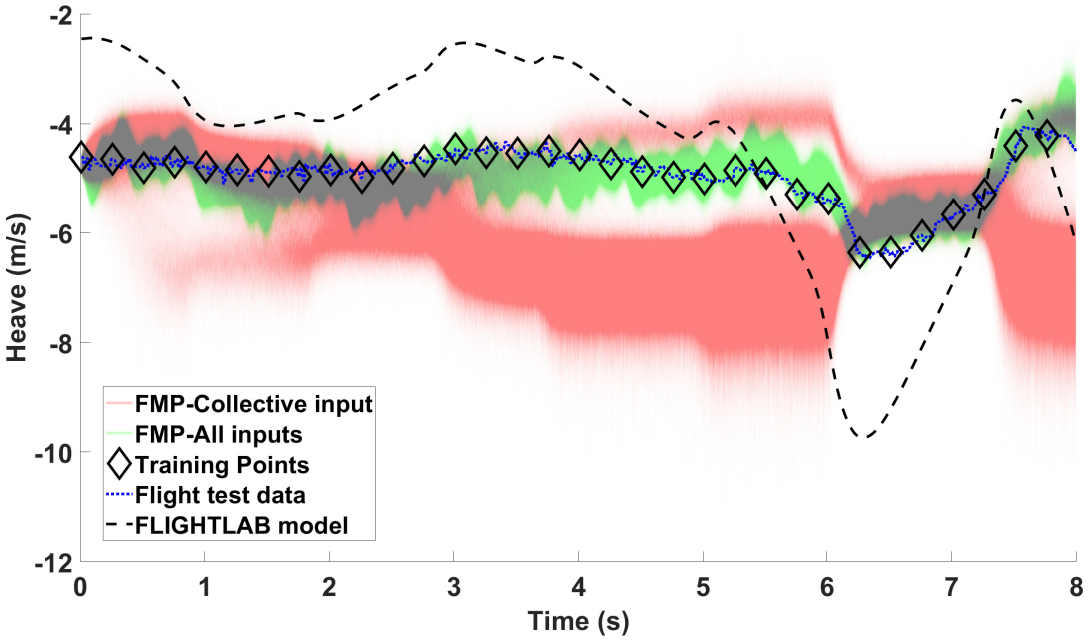


Figure 5.15: Realisations of the GP heave full model prediction for the ARX input structure containing the current collective lever pilot input with one lagged heave term and all of the current pilot inputs with one lagged heave term (cases 1 and 4 in Table 5.7) with comparison to the FLIGHTLAB model and corresponding flight test data.

For quantitative comparison, the similarity measure for the six cases for the heave response are displayed in Table 5.8. Using Table 5.8 and Figures 5.15, 5.16 and 5.17 it is clear to see that the ARX input structure that produced the most accurate heave FMP realisations utilises all of the

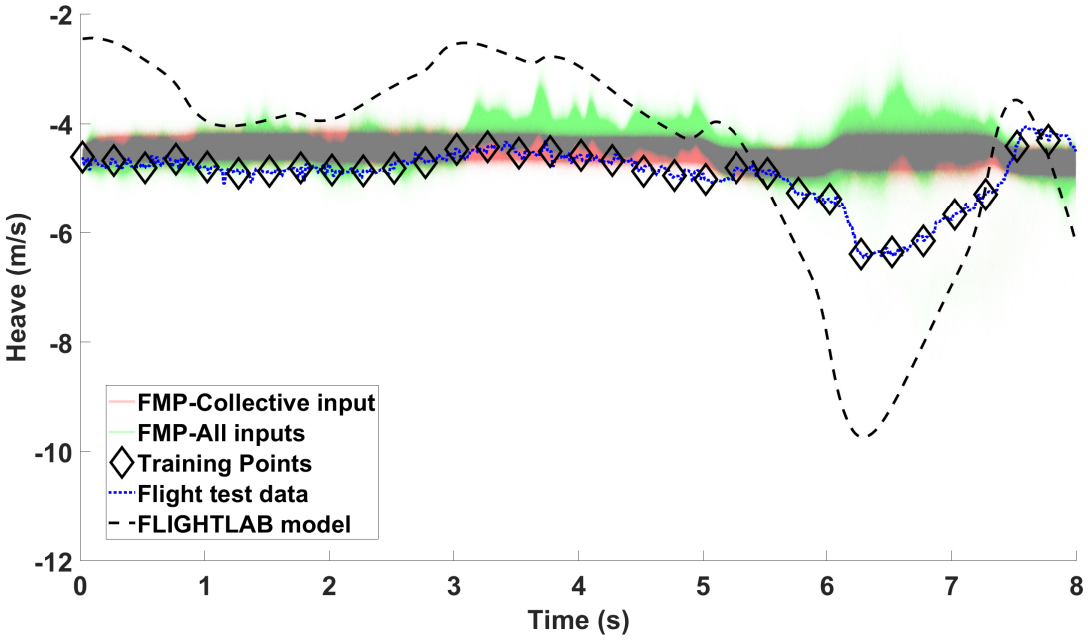


Figure 5.16: Realisations of the GP heave full model prediction for the ARX input structure containing the current collective lever pilot input with two lagged heave terms and all of the current pilot inputs with two lagged heave terms (cases 2 and 5 in Table 5.7) with comparison to the FLIGHTLAB model and corresponding flight test data.

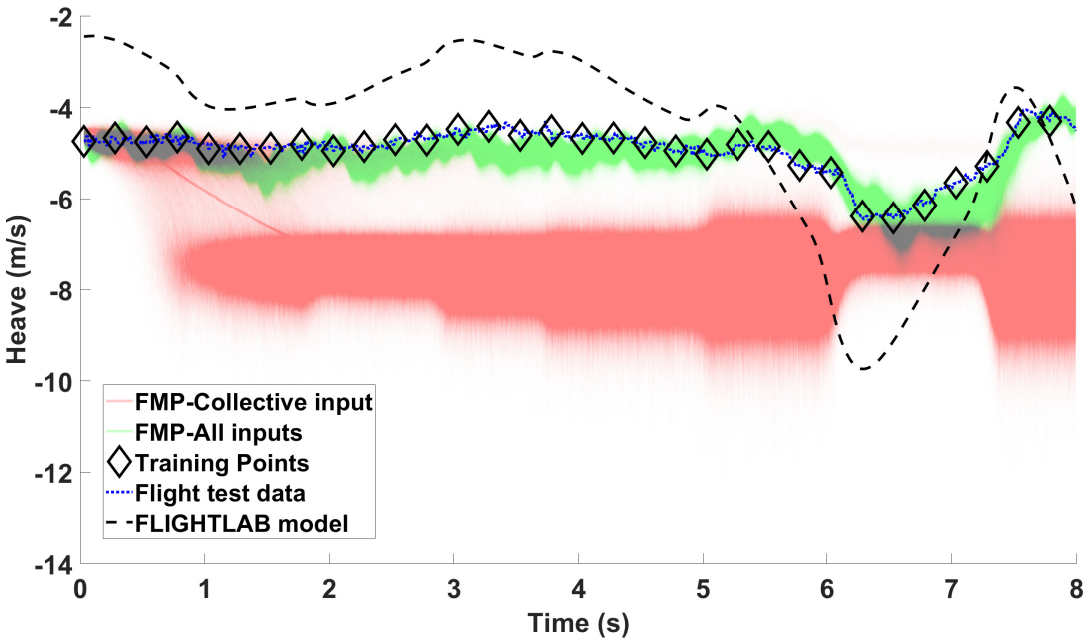


Figure 5.17: Realisations of the GP heave full model prediction for the ARX input structure containing the current collective lever pilot input with three lagged heave terms and all of the current pilot inputs with three lagged heave terms (cases 3 and 6 in Table 5.7) with comparison to the FLIGHTLAB model and corresponding flight test data.

current pilot inputs with one lagged heave term. It is worth noting that the heave is difficult to predict (e.g. the study by Punjani and Abbeel [4] struggled to predict the heave response) and the GP models display evidence for this, for example, when using the input structure containing all of the current pilot inputs, utilising two lagged heave terms produces results that are inferior compared to the one and three lagged heave terms. The heave predictions from the GP models are promising. However, the inconsistency of accuracy between the different input structures demonstrates that the heave response is difficult to predict. The most accurate input structure for heave does not follow the same pattern as the pitch rate, roll rate or yaw rate, in which two lagged response terms are required. This result is unexpected, as already discussed it is assumed that two lagged response terms are required to capture the underlying accelerations of the responses. Note that FLIGHTLAB predictions also struggled to capture the dynamics. The GP models outperform the FLIGHTLAB on 5 of the 6 cases, providing evidence that the GP can produce a more accurate initial model for the heave prediction than the FLIGHTLAB.

Prediction	**Current Inputs**	**Lagged Outputs**	**Case**	**Model**	**Similarity measure**
Heave	Collective input	One lagged heave term	1	OSA GP	1.46
				Mean FMP GP	505.44
		Two lagged heave terms	2	OSA GP	1.58
				Mean FMP GP	165.05
		Three lagged heave terms	3	OSA GP	1.41
				Mean FMP GP	1952.10
	Longitudinal, Lateral, Pedal and Collective inputs	One lagged heave term	4	OSA GP	1.86
				Mean FMP GP	**34.45**
		Two lagged heave terms	5	OSA GP	1.85
				Mean FMP GP	209.71
		Three lagged heave terms	6	OSA GP	1.68
				Mean FMP GP	42.80
				FLIGHTLAB	956.60

Table 5.8: The similarity measures of the different ARX input structures with varying pilot current inputs and lagged heave terms for the GP model predicting heave.

5.2 Comparison of Models

In Table 5.9, the 'best' models for each of the axes (pitch, roll, yaw and heave) are displayed. Two of the four models that produce the most accurate responses use two lagged response terms (pitch rate and yaw rate), and the roll rate models produced very similar results for all the input structures containing all of the current pilot inputs. This result is expected as the GP is attempting to predict a rate of change; therefore, the GP needs to take into account the acceleration of the response, using two lagged terms includes this information. The only exception is the heave GP model, in which the model using one lagged response term creates the most accurate model; this could be due to the difficulty of predicting the heave response.

5.2.1 Comparison of Accelerations

Up until this point, the similarity measure calculated the similarity between the mean of the GP and the flight test data. It is important to attempt to match the velocity and accelerations of the response and not just the velocity. In this section, the acceleration for the 'best' models for each axis (pitch rate, roll rate, yaw rate and heave) are plotted and compared to the accelerations fo the FLIGHTLAB model and flight test data.

Prediction	**Current Inputs**	**Lagged Outputs**	**Model**	**Similarity measure**
Pitch rate	Longitudinal input	Two lagged pitch rate terms	OSA GP	0.01
			Mean FMP GP	**3.52**
Roll rate	Longitudinal, Lateral, Pedal and Collective inputs	One lagged roll rate term	OSA GP	0.15
			Mean FMP GP	**10.30**
Yaw rate	Longitudinal, Lateral, Pedal and Collective inputs	Two lagged yaw rate terms	OSA GP	0.08
			Mean FMP GP	**53.26**
Heave	Longitudinal, Lateral, Pedal and Collective inputs	One lagged heave term	OSA GP	1.86
			Mean FMP GP	**34.45**

Table 5.9: The similarity measures for on-axis response GP models, shown in Section 5.1.1 for the comparison of the use of different GP input structures.

5.2.2 Pitch Acceleration

Firstly, the pitch acceleration is investigated. The input structure that produced the most accurate prediction was the current longitudinal pilot input with two lagged pitch rate terms. The acceleration realisations of the prediction are plotted in Figure 5.18, and for comparison, the FLIGHTLAB and flight test data accelerations are also plotted. It can be seen that the GP realisations encompass the majority of the flight test data and capture similar acceleration dynamics when compared to both the flight test data and FLIGHTLAB accelerations. The results from Figure 5.18 provide evidence that the GP predictions have similar accelerations to that of the flight test data. The only exceptions are after 6 seconds when the GP realisations go to 100 deg/s^2 and after 7 seconds when the GP realisations go to -100 deg/s^2, this could be due to missing information, for example, only using the longitudinal input and the not utilising the pilot's other inputs (lateral, pedal and collective). In the next paragraph, the accelerations for the GP model utilising the input structure containing all of the current pilot inputs (longitudinal, lateral, pedal and collective) is investigated.

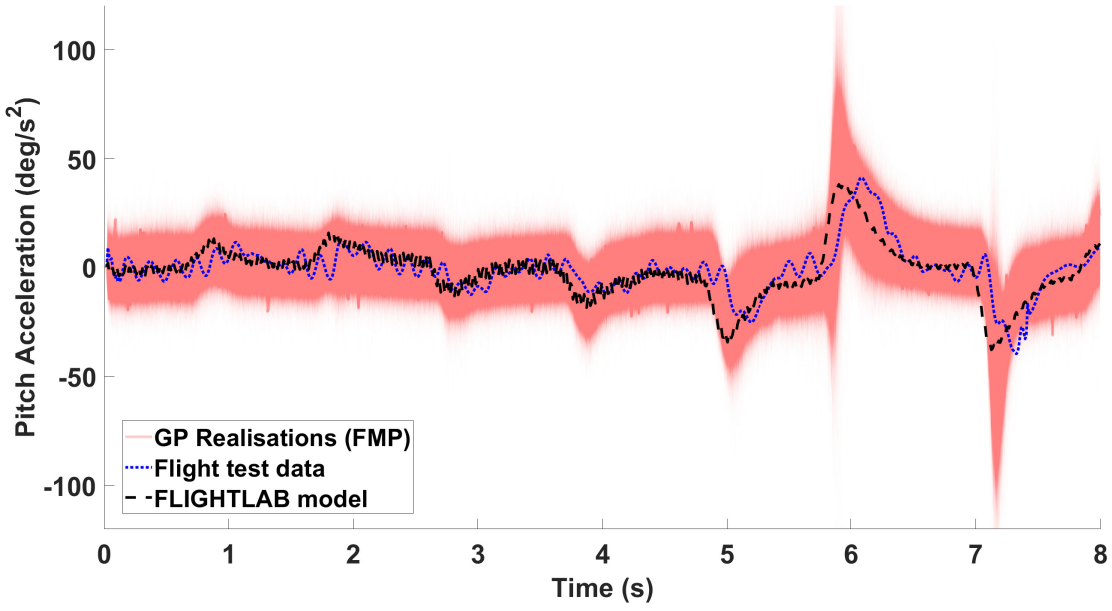


Figure 5.18: Realisations of the GP pitch rate full model prediction for the ARX input structure containing the current longitudinal pilot input with two lagged pitch rate terms and comparison to the FLIGHTLAB model acceleration and corresponding flight test data acceleration.

Figure 5.19 shows the acceleration for the pitch rate prediction from a GP using the input

structure of the all of the current pilot inputs (longitudinal, lateral, pedal and collective) and two lagged pitch rate terms. It is evident that the large accelerations (around 6 and 7 seconds) in Figure 5.18 are no longer as extreme in Figure 5.19. Comparing the GP realisations with FLIGHTLAB, the FLIGHTLAB accelerations are similar to the flight test data; however, the GP realisations are also very similar and follow the same dynamics. The result provides an intriguing conclusion, for the pitch rate, using the current longitudinal pilot input in the input structure provides a more accurate model in terms of comparison to the true pitch rate response. However, when observing the accelerations of the pitch rate response predictions, the model with the input structure containing all of the current pilot inputs is more accurate than the input structure containing the longitudinal pilot input. The model needs to capture the accelerations as well as the response. As this is the case, the input structure containing all of the current pilot inputs is used for the comparisons in subsequent chapters.

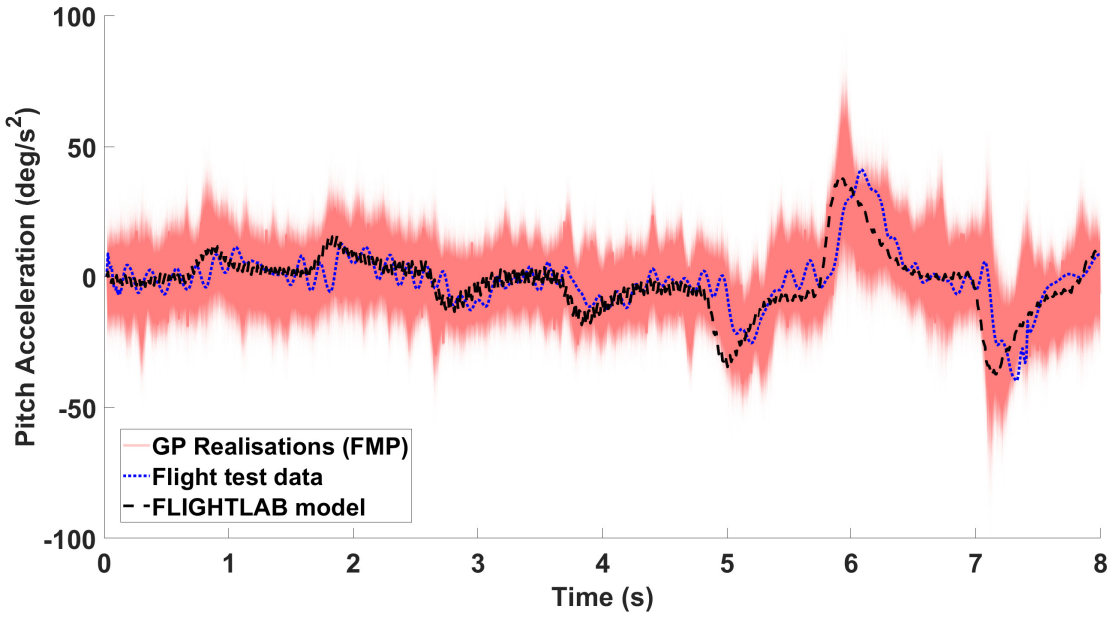


Figure 5.19: Realisations of the GP pitch rate full model prediction for the ARX input structure containing all of the current pilot inputs with two lagged pitch rate terms and comparison to the FLIGHTLAB model acceleration and corresponding flight test data acceleration.

5.2.3 Roll Acceleration

The most accurate input structure for the roll rate predictions contained all of the current pilot inputs with one lagged roll rate term. As discussed previously in section 5.1.2, the input structures containing all of the current pilot inputs produced very similar results. Figure 5.20

displays the acceleration for the roll rate predictions from the GP model containing all of the current pilot inputs and one lagged roll rate term. The acceleration realisations of the GP model are much larger when compared to the accelerations of the true rotorcraft. The reason for this may be due to only using one lagged roll rate term, the use of two lagged roll rate terms is investigated in the next paragraph.

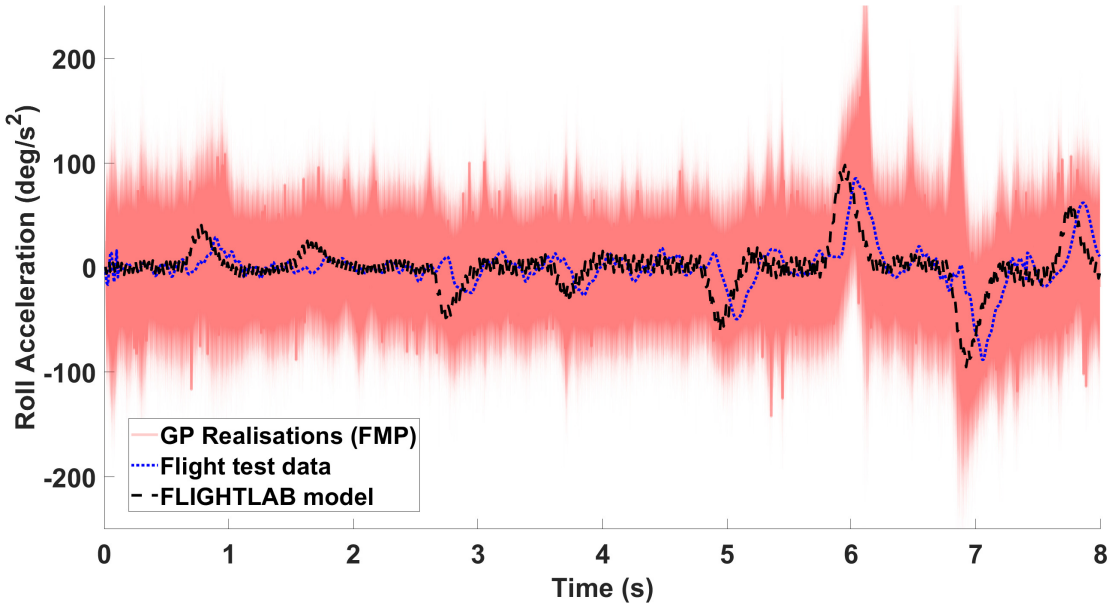


Figure 5.20: Realisations of the GP roll acceleration full model prediction for the ARX input structure containing all of the current pilot inputs with one lagged roll rate term and comparison to the FLIGHTLAB model acceleration and corresponding flight test data acceleration.

Figure 5.21 displays the acceleration for the roll rate predictions using the GP model, which has the input structure containing all of the current pilot current inputs and two lagged roll rate terms. Comparing Figures 5.20 and 5.21, the realisations in Figure 5.21 are much closer to the flight test data acceleration than Figure 5.20. The similarity measures for the roll rate GP predictions using all of the current pilot inputs are very similar. Therefore, for the subsequent chapters in the current thesis, the roll rate GP using the input structure containing all of the current pilot inputs and two lagged roll rate terms will be investigated.

Comparing the GP realisations with the FLIGHTLAB acceleration, it shows that both the FLIGHTLAB and GP realisations are similar to the flight test data. In some cases (time 5, 6 and 7 seconds) the FLIGHTLAB acceleration changes before the flight test data, the GP realisations match the spikes of the flight test data acceleration more accurately.

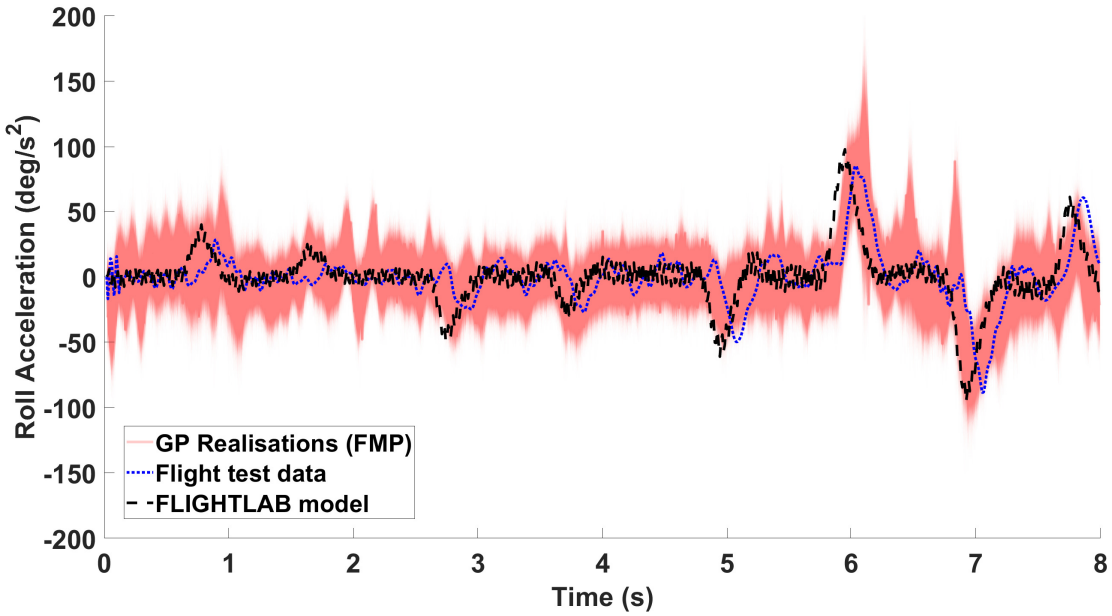


Figure 5.21: Realisations of the GP roll acceleration full model prediction for the ARX input structure containing all of the current pilot inputs with two lagged roll rate terms and comparison to the FLIGHTLAB model acceleration and corresponding flight test data acceleration.

5.2.4 Yaw Acceleration

The yaw acceleration for the GP using the input structure containing all of the current pilot inputs with two lagged yaw rate terms is displayed in Figure 5.22 along with the FLIGHTLAB and the flight test data accelerations. It can be seen that the GP acceleration realisations encompass all of the data, and there are visible spikes throughout the time series. However, unlike the previous two sections (pitch rate and roll accelerations) the most accurate yaw rate prediction is realised when utilising the input structure containing all of the current pilot inputs and two lagged yaw rate terms. Figure 5.22 displays evidence that for the majority of the manoeuvre, the acceleration is similar to that of the true rotorcraft, even with the visible spikes. Comparing the GP realisations acceleration with the FLIGHTLAB acceleration, the FLIGHTLAB is much smoother and matches the flight test data until 6 seconds after which it deviates.

5.2.5 Heave Acceleration

Figure 5.23 displays the heave acceleration from the GP model using the input structure containing all of the current pilot inputs and one lagged heave term, with comparison to the flight

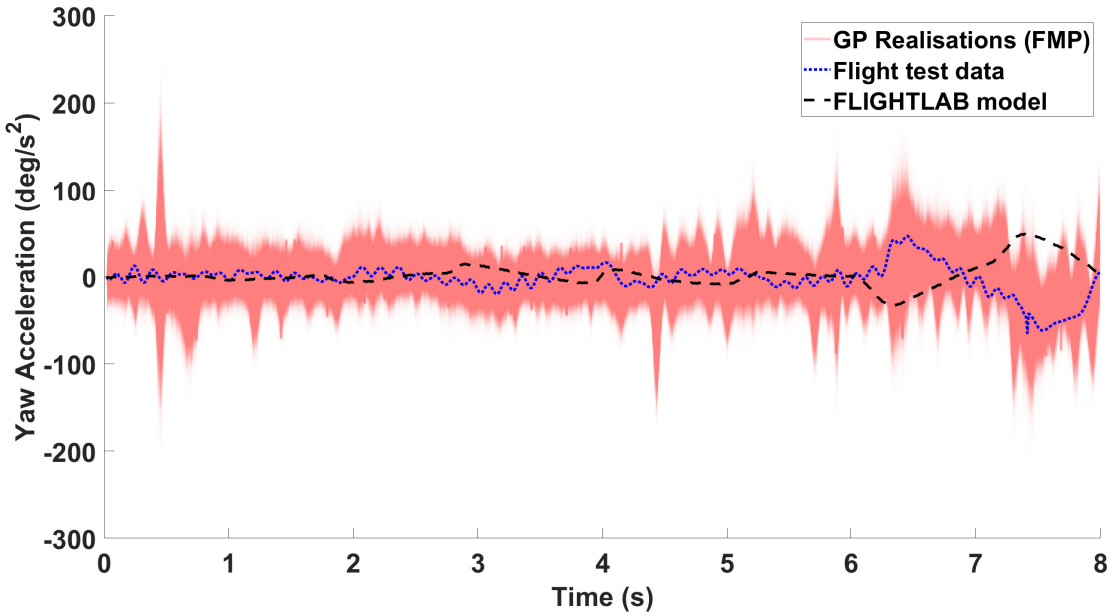


Figure 5.22: Realisations of the GP yaw acceleration full model prediction for the ARX input structure containing all of the current pilot inputs with two lagged yaw rate terms and comparison to the FLIGHTLAB model acceleration and corresponding flight test data acceleration.

test data and FLIGHTLAB model. Firstly, it is clear to see that the GP realisations span a broad range from roughly -50 to 50 m/s^2 and do not follow the dynamics of the flight test data; this could be due to using one lagged term in the input structure. However, the input structure containing two lagged responses and all of the current pilot inputs is not accurate (similarity measure for velocity is 209.71). Note that, as previously discussed, the GP struggles consistently to match the heave response.

5.3 Summary

The investigation of the use of different ARX input structures to improve the accuracy of the GP predictions was shown in the current chapter. Specifically, the use of all the current pilot inputs compared to the use of only the on-axis input of the response and the number of lagged response terms for the ARX input structure are investigated.

The GP results shown in the above sections are quantitatively summarised in Table 5.9 by the value of the similarity measure (discussed in section 4.1.6), displaying only the most accurate models. For the interested reader, the similarity measure for the OSAP are given in the Table; this provides evidence of the accuracy of the GP OSAP. It is clear to see that all of the GP

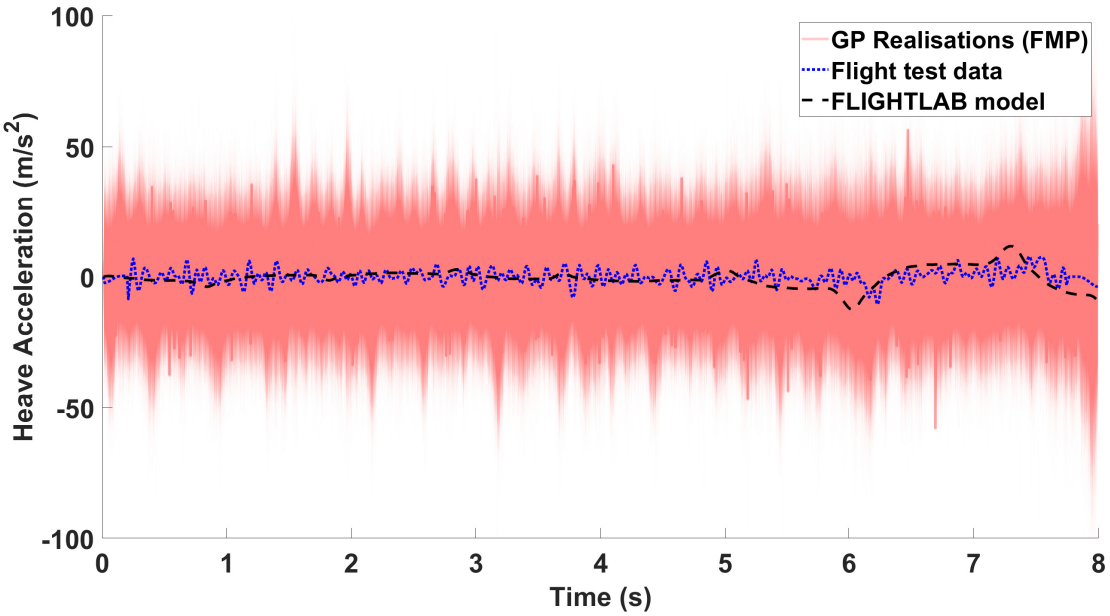


Figure 5.23: Realisations of the GP heave acceleration full model prediction for the ARX input structure containing all of the current pilot inputs with one lagged heave term and comparison to the FLIGHTLAB model acceleration and corresponding flight test data acceleration.

predictions outperform the FLIGHTLAB (note that the FLIGHTLAB model is calibrated by an expert user, discussed in Chapter 3).

The use of all the current pilot inputs for the ARX input structure improves the GP FMP realisations visually and quantitatively for the roll rate, yaw rate and heave. However, it does not improve the pitch rate; this could be due to the relatively stronger correlation between the longitudinal stick position to the response of the pitch rate. The use of all the current pilot inputs for the ARX input structure of the GP for the pitch rate response is very similar visually and quantitatively compared to only using the longitudinal input.

The number of lagged response terms used the ARX structure affected the accuracy of the model, for the pitch and yaw rate, the use of two lagged response terms produced more accurate results than using one lagged response term. As previously discussed, it is not only the accuracy of the prediction that is influential in creating an accurate model; it is important to investigate if the accelerations are similar. Using two lagged response terms for the roll rate provided predictions that were closer in terms of the acceleration.

The conclusions that have arisen from the current chapter are:

- For the pitch rate response, the FMP realisations for the GP with the ARX input structure containing the current longitudinal pilot input produce more accurate predictions than utilising all of the current pilot inputs for the ARX input structure. The outcome of the pitch rate response could be due to a strong correlation between the longitudinal stick position and the pitch rate. Another reason could be that the other inputs do not have that much of an effect on the response of pitch rate.

- The roll rate, yaw rate, and heave required the GP to utilise the ARX input structure of all of the current pilot inputs to capture the dynamics of the response accurately; this could be due to two reasons. The first reason could be due to the cross-coupling between the pilot's off-axis inputs and the response. The on-axis input may not be providing enough information for the GP to make accurate predictions. The second reason may be that using the ARX input structure of the current on-axis input with one lagged response term; there are only two GP inputs. During FMP the uncertain predicted output becomes a GP input. Therefore there is more reliance on the lagged response. The ARX input structure containing all of the current pilot inputs comprises of the four current pilot inputs with one lagged response term, which takes the reliance away from the uncertain lagged response. Also, when an incomplete input structure is utilised, in a relatively low-dimensional input space, the 'closeness' of new inputs to the training data may be erroneously estimated relative to if a higher dimensionality input space has been used (discussed in section 5.1.2).

- Using two lagged terms created more accurate velocity predictions for the pitch rate, and yaw rate response, this type of result seems reasonable as it was previously discussed that it is expected that the input structure containing two lagged terms would produce the most accurate results. As this would allow the model to use the second lag terms to capture the underlying accelerations of the response. Note, for the roll rate response the input structures containing all of the current pilot inputs created similar predictions.

- The acceleration of the prediction is an important aspect to consider, and it should be used along with the accuracy of the pitch rate response. The comparison of the

acceleration was demonstrated with the pitch and roll rate predictions in Sections 5.2.2 and 5.2.3, respectively. For the pitch rate response, the input structure containing all of the current pilot inputs produced more accurate results regarding the acceleration of the response compared the input structure containing only the current longitudinal pilot input. Concerning the roll acceleration, the input structure containing all of the current pilot inputs and two lagged roll rate terms produced more accurate realisations for acceleration than the input structure utilising all of the current pilot inputs and one lagged roll rate term.

For the rest of the current thesis, the ARX input structure containing all of the current pilot inputs and two lagged response terms will be used to predict the pitch, roll and yaw rate response, as this input structure has led to better predictions of acceleration and velocity. Note, that for the simulator to 'feel' like the true rotorcraft the underlying model needs to be able to predict the acceleration accurately. For the heave response, the input structure containing all of the current pilot inputs and one lagged heave term produced the most accurate predictions and will be used in the rest of the thesis. Note, as previously discussed, the heave response is difficult to predict.

Chapter 6

Investigation of Gaussian Process Hyperparameter Structure

In the previous chapter, different GP input configurations were investigated, with a combination of different pilot current inputs and the number of lagged response terms. In the current chapter, the use of different hyperparameter configurations is explored using the on-axis response; one investigates if the use of a hyperparameter for each input of the ARX structure improves the accuracy of the GP predictions. One employs Automatic Relevance Determination (ARD) to identify the most relevant inputs; this may indicate which pilot inputs are useful and may inform our understanding of the physical system. Assigning a hyperparameter to each input of the ARX input structure could also improve the GP FMP predictions. The chapter also explores potential issues around the hyperparameter tuning that is required in ARD implementations, as well as the generalisability of the proposed GP modelling approach to other manoeuvres.

6.1 Gaussian Process Hyperparameter Structure

In the current chapter, the ARX input structure containing all of the current pilot inputs with two lagged response terms is used for pitch rate, roll rate and yaw rate GP models. This is due to the results shown in the previous chapter, which concluded that all of the current pilot inputs were needed in the ARX input structure to capture the acceleration of the response. The exception to this is the heave, where using the ARX input structure containing the all of

the current pilot inputs with one lagged heave term produce more accurate predictions to that of the other combinations of ARX input structure.

In the previous chapter, four manoeuvres were considered, and the same manoeuvres are utilised in the current chapter. The manoeuvrers are summarised in Table 6.1. For each manoeuvre, two different hyperparameter structures are investigated. Each investigation is referred to as a 'case'. The first hyperparameter structure (cases 1, 3, 5 and 7, Table 6.1) uses one hyperparameter to govern the ARX input structure, and the kernel function is:

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \prod_{i=1}^{N_D} \alpha^{(4(\boldsymbol{x}_n^i - \boldsymbol{x}_m^i)^2)} \tag{6.1}$$

where x_i is used to represent the ith element of vector $\boldsymbol{x}$, N_D is the number of dimension of the input vector, n and m index the nth and mth input observation. For the pitch, roll and yaw rate, when using index n, $\boldsymbol{x_n}$ is:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \delta_n^y \\ \delta_n^p \\ \delta_n^o \\ y_{n-1} \\ y_{n-2} \end{pmatrix} \tag{6.2}$$

where δ_n^x is the longitudinal stick position, δ_n^y is the lateral stick position, δ_n^p is the pedal position, δ_n^o is the collective lever, and y_{n-1} is the lagged relevant observation (pitch, roll, yaw rate). The second hyperparameter structure (cases 2, 4, 6 and 8, Table 6.1) assigns a hyperparameter to each element of the ARX input structure, allowing the importance of each input to be determined. A hyperparameter value close to zero identifies the associated input as important, and for a hyperparameter value that is close to 1, the associated input is not important. The importance of the inputs can be used to inform which inputs do not have a significant effect on the prediction and can be removed for the input structure. When using the ARX input structure containing all of the current pilot inputs with two lagged response terms,

equation 6.2, the kernel function is

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \alpha_1^{4(\delta_n^x - \delta_m^x)^2} \times \alpha_2^{4(\delta_n^y - \delta_m^y)^2} \times \alpha_3^{4(\delta_n^p - \delta_m^p)^2} \times \alpha_4^{4(\delta_n^o - \delta_m^o)^2} \times \alpha_5^{4(y_{n-1} - y_{m-1})^2} \times \alpha_6^{4(y_{n-2} - y_{m-2})^2} \quad (6.3)$$

.

Manoeuvre	Prediction	Input structure	Number of Hyperparameter	Case
3-2-1-1 Longitudinal input	Pitch rate	All the current pilot inputs with two lagged pitch rate terms	One hyperparameter	1
			Six hyperparameters	2
3-2-1-1 Lateral input	Roll rate	All the current pilot inputs with two lagged roll rate terms	One hyperparameter	3
			Six hyperparameters	4
3-2-1-1 Pedal input	Yaw rate	All the current pilot inputs with two lagged yaw rate terms	One hyperparameter	5
			Six hyperparameters	6
3-2-1-1 Collective input	Heave	All the current pilot inputs with one lagged heave term	One hyperparameter	7
			Five hyperparameters	8

Table 6.1: GP configurations for the investigation of the GP hyperparameter structure

6.1.1 Pitch Rate

The OSAP for the GP using the ARX input structure containing all of the current pilot inputs with two lagged pitch rate terms and six hyperparameters are shown in Appendix C.1.

Figure 6.1 displays two sets (which are referred to as case 1 and 2 in Table 6.1) of GP FMP realisations using the ARX input structure containing all of the current pilot inputs with two lagged pitch rate terms. The difference between the two FMP realisation sets is the hyperparameter structure, the first set uses one hyperparameter (case 1, Table 6.1), shown in equation 6.1, and is denoted in Figure 6.1 by red realisations. The second set using six hyperparameters (case 2, Table 6.1), shown by equation 6.3, and displayed in Figure 6.1 by

green realisations. The hyperparameters used for the two cases are:

- Case 1 - All of the current pilot inputs with two lagged pitch rate terms using one hyperparameter - $\alpha = 0.992$ and $\beta = 5.946 \times 10^5$
- Case 2 - All of the current pilot inputs with two lagged pitch rate terms using six hyperparameters - $\alpha_1 = 0.999$, $\alpha_2 = 0.999$, $\alpha_3 = 0.999$, $\alpha_4 = 0.999$, $\alpha_5 = 0.961$, $\alpha_6 = 0.911$ and $\beta = 9.658 \times 10^5$

As with the GP FMP shown in chapter 5, Monte Carlo analysis is required for FMP predictions. Figure 6.1 displays two sets of predictions where both sets consist of an ensemble of Monte Carlo samples, each with 10,000 realisations. The Monte Carlo results are plotted using semi-transparent lines, such that darker regions represent areas where more realisations were observed. The grey realisations, displayed in Figure 6.1, are where the two sets of GP FMP realisations overlap. From observing Figure 6.1, the GP FMP realisations that use one hyperparameter (red realisations) outperform the GP FMP realisation using six hyperparameters (green realisations). Specifically, between time 5 to 7 seconds, the green realisations deviate from the flight test data. It is therefore evident in this case that using one hyperparameter is beneficial for the accuracy of the GP FMP realisations. The similarity measure between the mean FMP realisations and the flight test data for the GP using one hyperparameter is 5.52 and for the GP using six hyperparameters is 35.63. The similarity measures provide more evidence that the use of one hyperparameter, in this case, produces more accurate predictions. Even though the similarity measure for the GP using six hyperparameters is lower than the similarity measure for FLIGHTLAB prediction, it can be seen in Figure 6.1 that the GP using six hyperparameters (green realisations) appears to have multiple solutions. The trajectory taken during each Monte Carlo run seems to dictate which solution it converges to in the latter stages.

The hyperparameters values for case 2 are very similar; therefore, it is not apparent if any of the inputs are relevant/not relevant, and it is difficult to exclude any of the inputs. The hyperparameter result is not what is expected as, based on the results shown in Chapter 5, it is believed that the longitudinal stick position should have more importance.

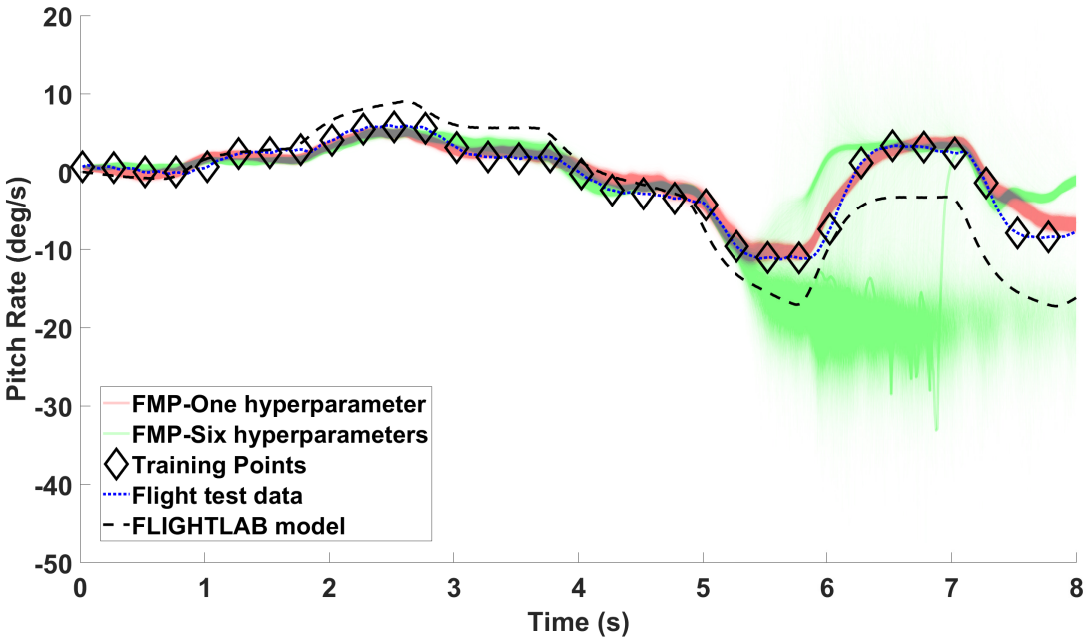


Figure 6.1: Realisations of the GP pitch rate full model prediction for the ARX input structure containing all of the current pilot inputs with two lagged pitch rate terms using one hyperparameter (case 1, Table 6.1) and six hyperparameters (case 2, Table 6.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

The convergence of the hyperparameters is investigated in more detail; the simulated annealing process was run four times (multiple runs to check the convergence) for GP model utilising the input structure of all of the current pilot inputs with two lagged pitch rate terms. The acceptance ratio for the final set of samples for the four simulated annealing run were, 30.00%, 38.95%, 29.37%, and 42.40%. Using the results from the acceptance ratios, the proposal is not poorly 'tuned'. The Markov Chains of log-likelihood values generated from the last set of samples from the four simulated annealing runs is shown in Figure 6.2; a different colour displays each simulated annealing run. It shows that for the majority of the time, the simulated annealing has converged, it does sometimes divert to another probably mass, as shown by the purple line at around 6000 samples.

Figure 6.3 displays Markov Chains for the six hyperparameters using the last set of samples from the simulated annealing optimisation for four runs. The simulated annealing runs are shown by different colours, which correspond to the same colours shown in Figure 6.2. Figure 6.3 supports the same result as shown in Figure 6.2, that it looks like the hyperparameters have converged and that the Markov Chain occasionally goes to another probably mass (blue

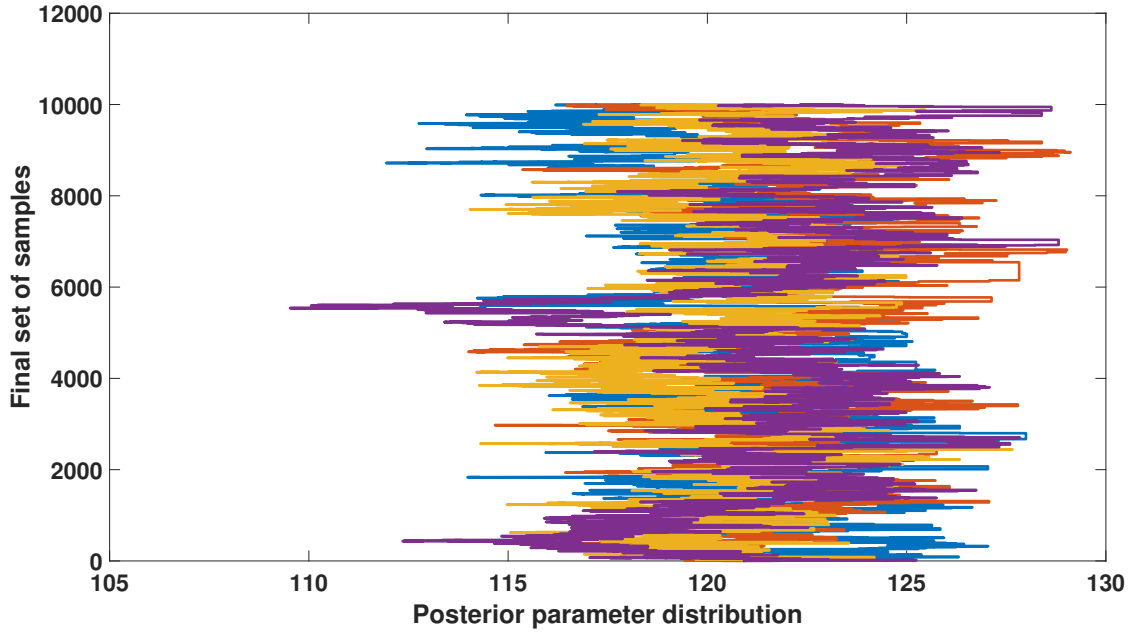


Figure 6.2: The log-likelihood values generated from the four simulated annealing results used to create the pitch rate GP models using the input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising six hyperparameters.

line in the subplot corresponding to hyperparameter (α_6)). The fact that the Markov Chain does divert to other areas suggests that the posterior is relatively complex, i.e. there are local minima. The complex nature of the posterior could be the reason why the FMP for the GP model utilising six hyperparameters is not as accurate as when one hyperparameter is used. Another possible hypothesis is that assigning a hyperparameter to each input in the input structure gives the model added 'flexibility', in which, it is overfitting. The GP model could be finding a correlation between the input and output that is valid over the training data but does not hold over the whole dataset, which would be exacerbated by the full model predictions. It is also worth noting, that a hyperparameter value close to one indicates that the corresponding input is not relevant to the prediction, in the current case, using the recorded results, four of the hyperparameters are very close to 1 they were 0.999. It has already been stated that all of the hyperparameters are similar, which they are all greater than 0.91. If we indicate the four input corresponding to the hyperparameters that have a value of 0.999 to be not relevant, this would include the four pilot current inputs ($\delta_n^x, \delta_n^y, \delta_n^p$ and δ_n^o). Which would leave the two lagged pitch rate terms as the relevant inputs to the prediction, which in reality does not make much sense, as the pilots inputs are crucial to the response to of the rotorcraft. It has already

been discussed in Chapter 5, the importance of the longitudinal stick position to the pitch rate prediction. Which could be an indication that the optimisation process has not converged to the global minima.

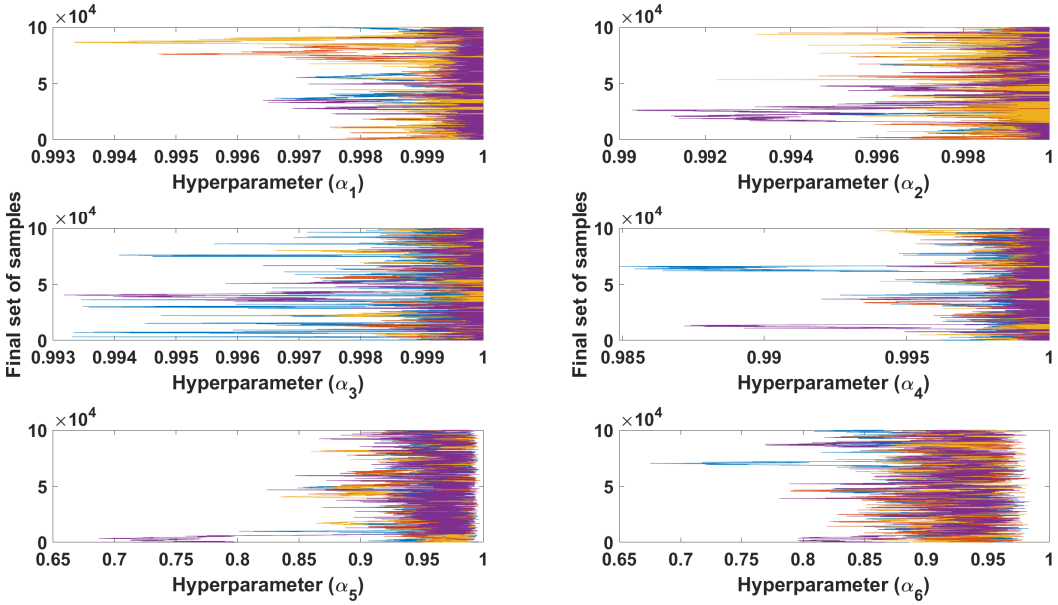


Figure 6.3: Markov chains for the four simulated annealing results for the six hyperparameters used to create the pitch rate GP models using the input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising six hyperparameters.

6.1.2 Roll Rate

The roll rate on-axis response is investigated in the current section. The OSAP and simulated annealing results corresponding to the GPs shown in this section are given in Appendix C.2. Figure 6.4 shows two sets of GP FMP realisation for the roll rate response using the ARX input structure containing all of the current pilot inputs with two lagged roll rate terms. The first set of GP FMP realisations is when the GP used one hyperparameter (case 3, Table 6.1) displayed by red realisations in Figure 6.4. The kernel structure using one hyperparameter is shown in equation 6.1. The second set of GP FMP realisations is where the GP used six hyperparameters (case 4, Table 6.1) displayed in Figure 6.4 as green realisations. The hyperparameters used for the two sets were:

- Case 3 - All of the current pilot inputs with two lagged roll rate terms using one hyperparameter - $\alpha = 0.992$ and $\beta = 4.602 \times 10^4$

- Case 4 - All of the current pilot inputs with two lagged roll rate terms using six hyperparameters - $\alpha_1 = 0.997$, $\alpha_2 = 0.995$, $\alpha_3 = 0.995$, $\alpha_4 = 0.995$, $\alpha_5 = 0.983$, $\alpha_6 = 0.950$ and $\beta = 5.177 \times 10^4$

The hyperparameters for case 4 are all very similar; therefore, one can not conclude that any of the inputs are relevant/not relevant. The result is similar to what is shown for the pitch rate predictions.

Visually, using Figure 6.4, it is not apparent which ensemble of GP FMP realisations are closest to the flight test data. Using the similarity measure between the mean FMP realisations and the flight test data for the GP using one hyperparameter is 10.83 and for the GP using six hyperparameters is 15.09. Quantitatively, it is shown that there is very little difference between using one or six hyperparameters to govern the roll rate GP. The reason why the GP model that utilised six hyperparameters is not as accurate as the GP model that used one hyperparameter is discussed in Section 6.1.1. In this case, it may therefore be favourable to use one hyperparameter to reduce the complexity of the optimisation search area. However, the GP FMP realisations outperform the FLIGHTLAB model in both cases.

6.1.3 Yaw Rate

The on-axis yaw rate response is explored in the current section. Two ensembles of GP FMP realisations are shown in Figure 6.5. The GP FMP realisations utilise the ARX input structure containing all of the current pilot inputs with two lagged yaw rate terms. The first ensemble of realisations uses the hyperparameter structure containing one hyperparameter (case 5, Table 6.1) shown in Figure 6.5 as red realisations. The second uses the hyperparameter structure containing six hyperparameters (case 6, Table 6.1) and is shown as green realisations in Figure 6.5. The hyperparameter for the two cases for the yaw rate response are:

- Case 5 - All of the current pilot inputs with two lagged yaw rate terms with one hyperparameter - $\alpha = 0.991$ and $\beta = 7.222 \times 10^4$
- Case 6 - All of the current pilot inputs with two lagged yaw rate terms with six hyperparameters - $\alpha_1 = 0.999$, $\alpha_2 = 0.999$, $\alpha_3 = 0.999$, $\alpha_4 = 0.999$, $\alpha_5 = 0.951$, $\alpha_6 = 0.905$ and

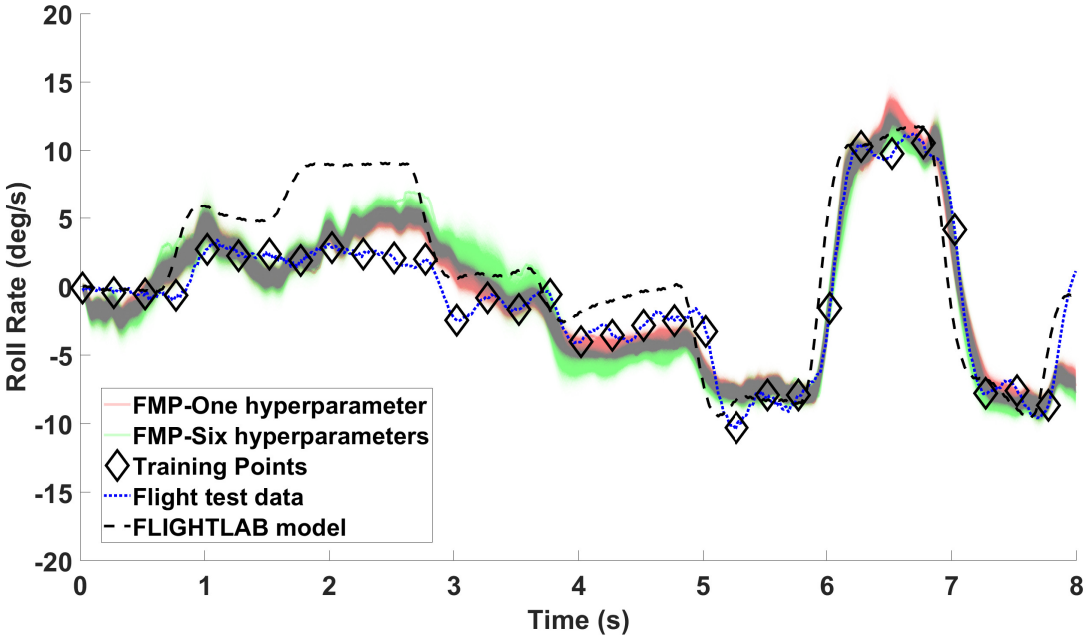


Figure 6.4: Realisations of the GP roll rate full model prediction for the ARX input structure containing all of the pilot current inputs with two lagged roll rate terms using one hyperparameter (case 3, Table 6.1) and six hyperparameters (case 4, Table 6.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

$$\beta = 4.191 \times 10^{6}$$

As with the previous two sections, pitch and roll rate, the hyperparameter structure when using a hyperparameter assigned to each input in the ARX input structure are all very similar. It is not clear if any of the inputs are more relevant than the others. For the interested reader, the OSAP and the simulated annealing results for the GP figures shown in this section are displayed in Appendix C.3.

It is evident from Figure 6.5 that the GP FMP realisations that using one hyperparameter (red realisations) produce more accurate results than using six hyperparameters (green realisations). The similarity measure between the mean FMP realisations and the flight test data for the GP using one hyperparameter is 53.26 and for the GP using six hyperparameters is 154.40, which confirms what is visible in Figure 6.5. It is therefore clear that in the case of the yaw rate response, the use of one hyperparameter produces more accurate results. The use of six hyperparameters should, in theory, improve the GP predictions, as the method assigns a hyperparameter for each GP input in the ARX input structure. Using six hyperparameters

allows for more flexibility due to the different hyperparameters for each input compared to using one hyperparameter to govern the ARX input structure. As already discussed the extra flexibility for the ARD GP model could be causing the model to overfit to the training data.

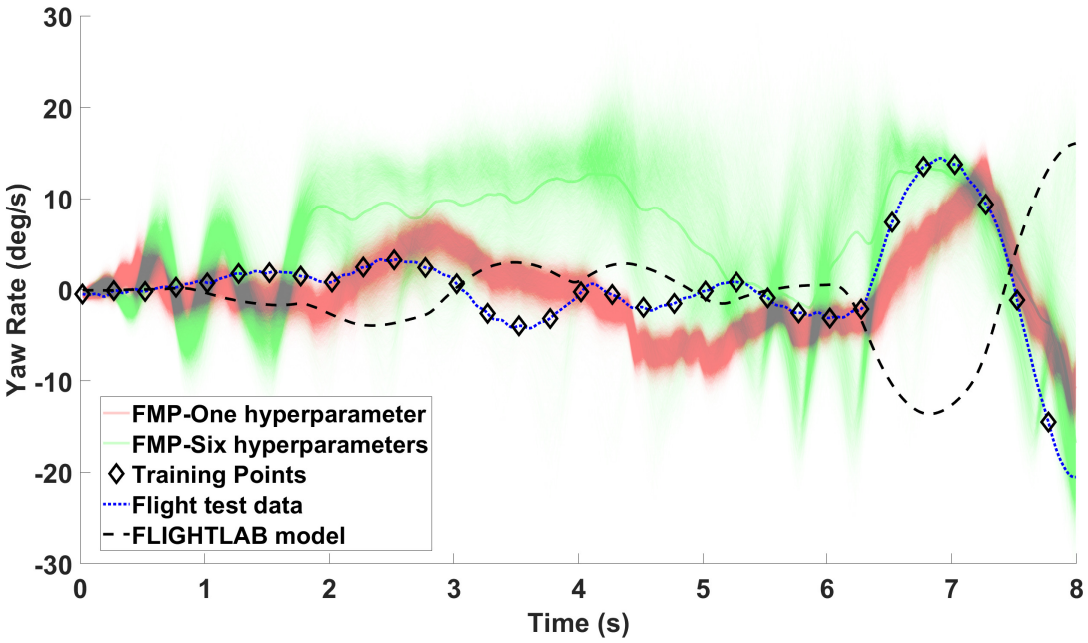


Figure 6.5: Realisations of the GP yaw rate full model prediction for the ARX input structure containing all of the pilot current inputs with two lagged yaw rate terms using one hyperparameter (case 5, Table 6.1) and six hyperparameters (case 6, Table 6.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

6.1.4 Heave

The on-axis heave response is investigated in the current section. The OSAP and simulated annealing results are not shown in the current section; however, it can be seen in Appendix C.4.

Figure 6.6 displays two sets of Monte Carlo analysis for GP FMP. Both sets use the ARX input structure containing all of the current pilot inputs with one lagged heave term; however, the hyperparameter structure differs. The first set of GP FMP realisations uses one hyperparameter (case 7, Table) displayed in Figure 6.6 as red realisations. The kernel structure using one hyperparameter is shown in equation 6.1. The second set of GP FMP realisations use five hyperparameters (case 8, Table) and displayed in Figure 6.6 by green realisations. As with previous GP FMP figures in the current thesis, the grey area represents an overlap of the two

sets of realisations. The hyperparameters for the two cases shown in the current section are:

- Case 7 - All of the current pilot inputs with one lagged heave term using one hyperparameter - $\alpha = 0.986$ and $\beta = 1.192 \times 10^4$
- Case 8 - All of the current pilot inputs with one lagged heave term using five hyperparameters - $\alpha_1 = 0.989$, $\alpha_2 = 0.982$, $\alpha_3 = 0.951$, $\alpha_4 = 0.994$, $\alpha_5 = 0.898$ and $\beta = 1.394 \times 10^3$

The hyperparameters for case 8 are all very similar; therefore, it is currently unclear if any of the GP inputs are more relevant than others.

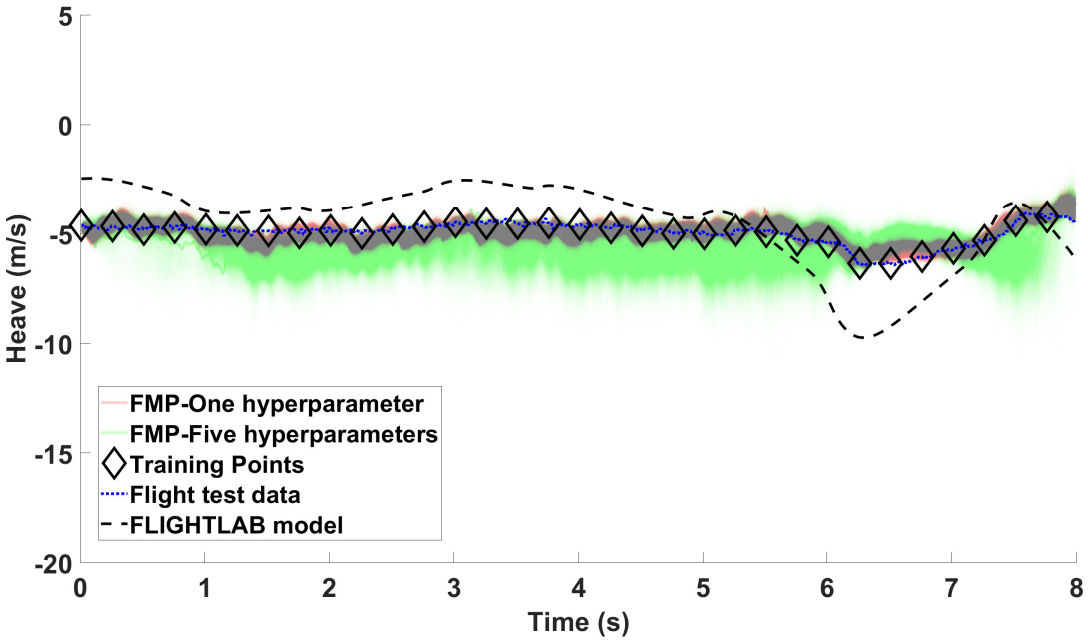


Figure 6.6: Realisations of the GP heave full model prediction for the ARX input structure containing all of the current pilot inputs with one lagged heave term using one hyperparameter (case 7, Table 6.1) and five hyperparameters (case 8, Table 6.1) with comparison to the FLIGHTLAB model and corresponding flight test data.

Figure 6.6 shows that the red realisations (one hyperparameter) produced more accurate results than the green realisations (five hyperparameters). It is, therefore, more beneficial to use one hyperparameter for the on-axis heave response. The similarity measure between the mean FMP realisations and the flight test data for the GP using one hyperparameter is 34.45 and for the GP using five hyperparameters is 131.61. The similarity measure values confirm what is easily observable from Figure 6.6, that using one hyperparameter produces more accurate predictions.

6.2 Generalisation of the Gaussian Process Models

In this section, investigations are performed to see if the GP models can be applied manoeuvres outside of the training set. The model displayed in the current section is trained on the 3-2-1-1 longitudinal input to predict pitch rate using the best performing input structure, which was the input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising one hyperparameter (from results displayed in Chapter 5 and the current chapter). The GP model is used to predict the pitch rate of the doublet manoeuvre. Using results from the previous chapters, the pitch rate GP predictions for the 3-2-1-1 manoeuvre is the most accurate compared to the GP predictions for the roll rate, yaw rate, and heave. The pitch rate doublet prediction using the model trained on the 3-2-1-1 manoeuvre is investigated first, with the aim that if the pitch rate doublet predictions are accurate one would investigate the roll rate, yaw rate and heave. Note that only the FMP realisations are displayed in the section.

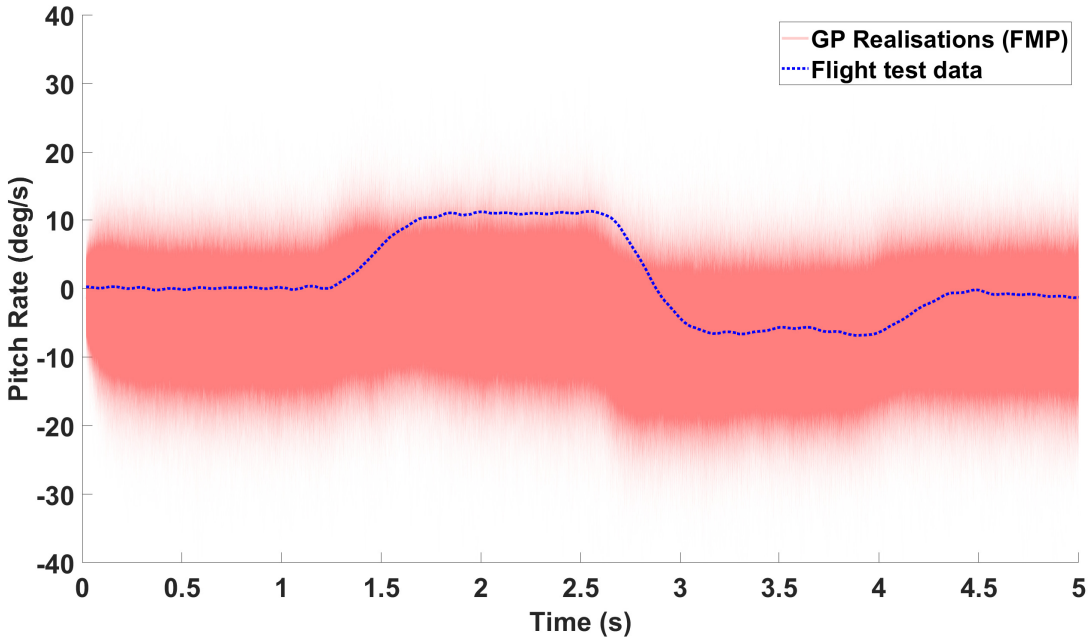


Figure 6.7: Realisations of the GP pitch rate full model prediction for the validation data (doublet response) using ARX input structure containing all of the current pilot inputs with two lagged pitch rate terms with comparison to the flight test data.

Figure 6.7 displays FMP realisations for the pitch rate doublet manoeuvre using the model trained on the pitch rate 3-2-1-1 manoeuvre. The predictions struggle to capture the dynamics of the doublet response, and the GP is very uncertain when making the prediction as the

realisation bounds are very large. The similarity measure for the mean FMP realisations and the flight test data is 170.75; therefore, the model does not generalise adequately to data from a different manoeuvre. It has been shown that the pitch rate prediction for the 3-2-1-1 manoeuvre is relatively accurate; however, the results in Figure 6.7 show that the model is poor at extrapolating to new manoeuvres, which is a potential problem if the model is was to be used in a flight simulator. The poor extrapolation can be reduced by using training data that covers a greater range of manoeuvres; however, it is known that GPs have a significant computational cost with large datasets. Therefore, it is beneficial to reduce the number of training points, which involves selecting the most informative points from the dataset, this is investigated in Chapters 7 and 8, where the performance of sparse GPs is explored. The sparse GP enables the use of larger datasets as the method attempts to select the most informative points and therefore reduces the associated computation cost; potentially allowing future implementations to be trained on datasets that contain multiple manoeuvres.

6.3 Summary

The use of different hyperparameter structures is considered in the current chapter, where the two cases are for one hyperparameter and different hyperparameters assigned to each input in the ARX structure. The ARX input structure utilising one hyperparameter produces more accurate results for the pitch rate, roll rate, yaw rate, and heave.

Table 6.2 contains the similarity measure for all the cases shown in the current chapter. The similarity measure confirms what is evident in the figures, that the use of one hyperparameter provides more accurate predictions for all of the responses, with the roll rate producing close results. In theory, the use of more hyperparameters should improve the accuracy of the GP predictions as it allows for more flexibility. The assignment of a hyperparameter for each GP input should allow the inputs with great importance on the response to have a significant effect on the predictions. As discussed in Section 6.1.1, the added flexibility allowed by assigning a hyperparameter to each input could be causing the GP model to overfit, this would create a model that is not good at generalising beyond the trained data. This would be the case when the GP uses full model predictions, as the prediction becomes an input and the uncertainty

propagates. One solution could be to use a different training set; using extra data covering a more comprehensive range of manoeuvrers would change the search space, and could potentially solve the issue. Using a different training set is not undertaken in the current work but could be addressed in the future.

In the next chapters, the use of one hyperparameter will be used as in the current chapter, as assigning a hyperparameter to each input of the ARX structure did not improve the predictions.

Prediction	**Number of hyperparameters**	**Input structure**	**Case**	**Model**	**Similarity measure**
Pitch rate	One hyperparameter	All of the current pilot inputs with two lagged pitch rate terms	1	OSAP GP	**0.03**
				Mean FMP GP	**5.52**
	Six hyperparameters		2	OSA GP	**0.01**
				Mean FMP GP	**35.63**
				FLIGHTLAB	**75.02**
Roll rate	One hyperparameter	All of the current pilot inputs with two lagged roll rate terms	3	OSAP GP	**0.15**
				Mean FMP GP	**10.83**
	Six hyperparameters		4	OSA GP	**0.11**
				Mean FMP GP	**15.09**
				FLIGHTLAB	**42.43**
Yaw rate	One hyperparameter	All of the current pilot inputs with two lagged yaw rate terms	5	OSAP GP	**0.08**
				Mean FMP GP	**53.26**
	Six hyperparameters		6	OSA GP	**0.01**
				Mean FMP GP	**154.40**
				FLIGHTLAB	**370.08**
Heave	One hyperparameter	All of the current pilot inputs with one lagged heave term	7	OSAP GP	**1.86**
				Mean FMP GP	**34.45**
	Five hyperparameters		8	OSA GP	**1.69**
				Mean FMP GP	**131.61**
				FLIGHTLAB	**956.60**

Table 6.2: The similarity measure for on-axis pitch rate response GP models, shown in Section 6.1 for the comparison of the use of different GP hyperparameter structures.

Regarding generalisation, the GP model was poor at generalising to the doublet manoeuvre; this could be due to the training input (3-2-1-1 data) being far from the prediction input

(doublet data). If this is the case, the GP will struggle to accurately predict the response as the GP has not 'seen' the scenario before. The generalisation could be improved by using data from different manoeuvres to train the model (note that this data would typically be available during a flight test campaign).

Chapter 7

Background Theory for the Variational Sparse Gaussian Process

In the current chapter, the sparse GP is introduced. Firstly some background to the sparse GP and the associated notable literature is provided before the detailed methodology is expanded upon. The methodology displayed in the current chapter was introduced by Titsias [17, 18] and is presented in detail to create a self-contained document. The author implemented the existing algorithm to create the results in Chapters 8 and 9.

7.1 Sparse Gaussian Process Background

The creation of a model for a dynamical system requires a set of input/output data; the data is used to establish a relationship between the input and output. The amount of data required to allow for accurate predictions will vary depending on the complexity of the system. Selecting the 'correct' amount of training data is crucial for the creation of an efficient and accurate model. The use of too much data will produce a model that will take longer than necessary to train and generate predictions. Too little data will have the opposite effect; the optimisation and predictions will be quicker; however, the accuracy of the model will be compromised. Using too much data is preferred over too little as an inaccurate model has little use.

In the case of the current thesis, the model that is used is a GP. The application of GP models

are intractable for large data sets because of the time complexity scale of $O(N^3)$, where N is the number of training points [17, 18]. A reduction in the number of training points is therefore beneficial. A GP model that has been created with a smaller subset of the original set of training points is referred to as a sparse GP. The aim is to make the sparse GP model predictions as similar as possible to the predictions made by the GP model trained on all the training data. Note that the sparse GP could outperform the full GP, as the data points omitted may be noise or not a true representation of the underlying system.

The concept of attempting to reduce the number of training points for GPs is not a new idea, and there have been many approximate/sparse methods that have been proposed, and examples of some literature are

- A unifying view of the sparse approximation is given by Quinonero-Candela and Rasmussen [63], in which many other methods of sparse approximations for GP regression are reviewed and ranked by how close the approximation is to the full GP. All the methods reviewed reduces the computational cost to $O(N^2M)$, where M is the number of sparse points and $M < N$, apart from the 'subset of data approach' which is $O(M^3)$ (where one selects M sparse points in an ad-hoc manner).

- Smola and Bartlett [64] use a sparse greedy technique to approximate the likelihood and maximise the posterior by using points from the original dataset (creating a subset from the original dataset), reducing the computational cost to $O(N^2M)$, where M is the smaller subset of data.

- The sparse points do not always have to be in the original dataset, as shown by Snelson and Ghahramani [65] used pseudo-inputs (inputs that are not part of the original data). The authors proposed to approximate the likelihood. The advantage of this method is that the data points are not constricted to the original dataset, which, it is shown, can lead to better solutions. However, the method encountered problems for very large dataset/high dimensional input space using the gradient based-optimisation.

- A sparse greedy approach is employed by Seeger et al. [66], similar to that used in [64], which attempts to approximate the likelihood and use what is referred to as the maximum

information gain. However, as stated by Titsias [17, 18], the objective function derived does not monotonically increase with the number of sparse points.

- Previous work to create a sparse GP has used different variational distributions such as projected process approximation (PP) [66] and sparse pseudo-input GPs (SPGPs) [65]. The two methods use different approaches to approximate the log marginal likelihood; however, both methods use an approximate of the log-likelihood of the form:

$$F = \log \left[\mathcal{N}(y \mid \mathbf{0}, \sigma^2 \boldsymbol{I} + \boldsymbol{A}_{NN}) \right] \tag{7.1}$$

  where $\boldsymbol{A}_{NN}$ is an approximation of the true covariance matrix $\boldsymbol{K}_{NN}$. Both approximation methods [65, 66] have a computational cost of $O(NM^2)$.

- Williams and Seeger [67] use a Nyström approximation to create a low-rank covariance matrix. One disadvantage of this method is that the approximate covariance matrix is not guaranteed to be positive definite; therefore, the predicted variance is not guaranteed to be positive. The computational cost of this method is $O(NM^2)$.

All the above methods attempt to find a reduced training set. One reason the variational sparse GP was used as it actively attempts to minimise the difference between the sparse GP and a GP trained on all the available data. Sometimes in the literature, the optimal training points are referred to as 'inducing points'. However, in the current chapter, the optimal training points will be referred to as sparse points.

To illustrate examples of good and bad choices for the sparse points, a toy example is used for demonstration purposes in Figures 7.1 and 7.2. These illustrate cases where the sparse points are chosen manually. The sparse GP models in Figures 7.1 and 7.2 are fitted to data generated from a sine wave without any measurement noise. Figure 7.1 displays the effect of choosing some 'non-optimal' sparse points. An example of the possible training points are indicated as diamonds. The five training points are located to the left of the figure and are close together. Due to the choice of the training points, the model performance is negatively affected from 3 radians onwards; this is displayed by the uncertainties shown by the confidence

bounds (expanding bounds).

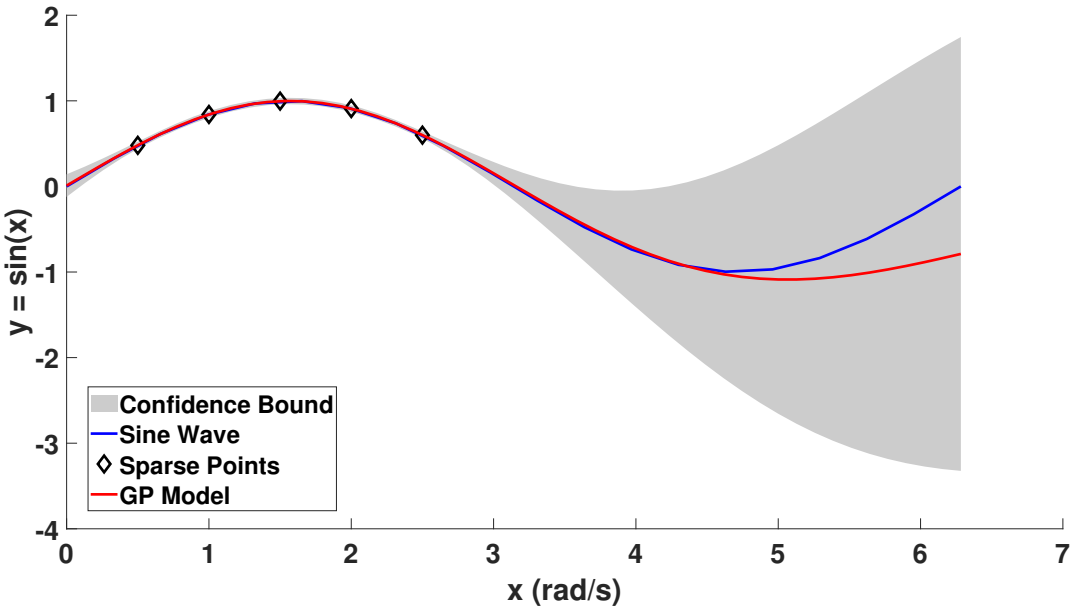


Figure 7.1: Manually selected training points creating a Sparse GP with a poor fit.

The 'ideal' sparse GP will replicate what the 'full' GP will have created given all the training data. A better performing sparse GP is shown in Figure 7.2; the GP is created only using five sparse points. The locations of the sparse points are more spread out and in important locations (at the crest and trough of the sine wave). The locations of the sparse points were chosen manually as it is easy to see where the optimal locations for the points are in a sine wave. Manual selection is, however, impractical when it is not clear where the most informative training points will be located, which, for most real systems, will be the case. An approach for identifying the optimal locations for the sparse points will be discussed in more detail in the subsequent sections. As previously discussed, the confidence bounds can indicate where more training data is required and can be utilised in the GP model development. In Figure 7.1 it is easy to see that training data is required from 3 radians onwards to accurately predict the sine wave and this is in confirmed in Figure 7.2, where training data is even spaced across the time-series. The sine wave example displays how the confidence bounds can be used in the GP model development (which is discussed in more detail in 10.2).

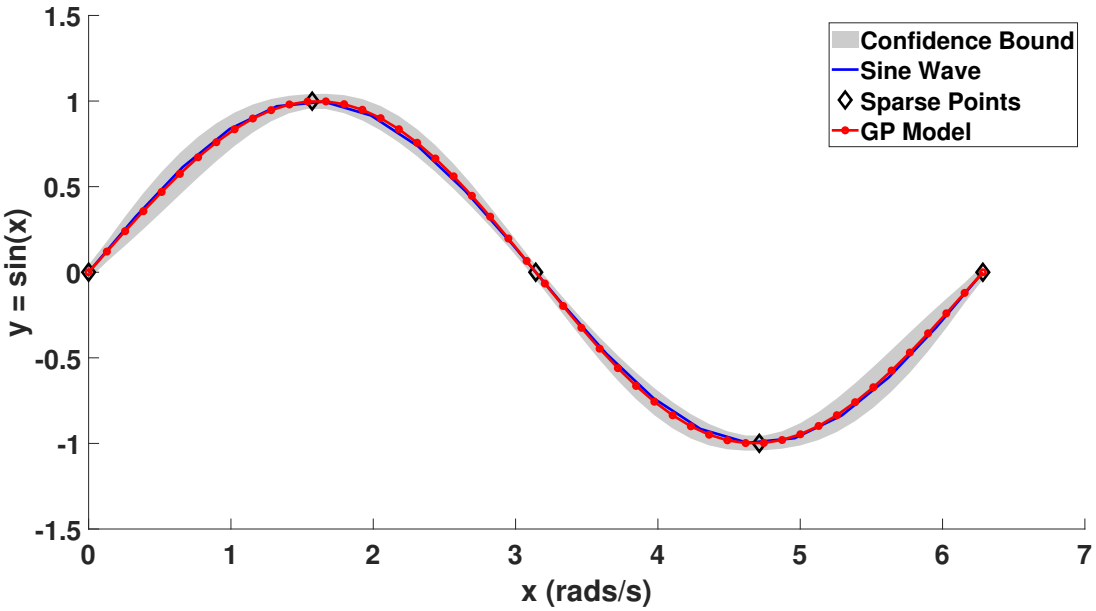


Figure 7.2: Manually selected training points creating a Sparse GP with a good fit.

7.2 Sparse Gaussian Process Theory

As previously discussed, the main reason to use a sparse GP is to reduce the number of training points and allow for improved efficiency. In the current chapter, a method called a variational sparse GP [17, 18] is introduced. As already stated, the theory displayed in the current chapter is not novel from the author, but the approach is expanded upon here in the interest of creating a self-contained document. The implemented methodology produces the sparse GP results in the next chapters.

7.2.1 Introduction to the Variational Bayes Framework

Firstly, a brief introduction to Variational Bayes is given to aid clarity. The goal of the variational method is to find an approximation for the posterior distribution $p(\boldsymbol{V} \mid \boldsymbol{y})$, where $\boldsymbol{V}$ is all the latent variables and parameters ('hidden' functions and parameters) of a particular problem and $\boldsymbol{y}$ are observations. The KL divergence can be written as

$$\mathrm{KL}(q \mid\mid p) = -\int q(\boldsymbol{V}) \log \left\{ \frac{p(\boldsymbol{V} \mid \boldsymbol{y})}{q(\boldsymbol{V})} \right\} d\boldsymbol{V}. \tag{7.2}$$

It can then be shown that

$$\log(p(\boldsymbol{y})) = \mathcal{L}(q) + \mathrm{KL}(q \,||\, p) \tag{7.3}$$

where

$$\mathcal{L}(q) = \int q(\boldsymbol{V}) \log \left\{ \frac{p(\boldsymbol{y}, \boldsymbol{V})}{q(\boldsymbol{V})} \right\} d\boldsymbol{V}$$

and therefore

$$\log(p(\boldsymbol{y})) = \int q(\boldsymbol{V}) \log \left\{ \frac{p(\boldsymbol{y}, \boldsymbol{V})}{q(\boldsymbol{V})} \right\} d\boldsymbol{V} - \int q(\boldsymbol{V}) \log \left\{ \frac{p(\boldsymbol{V} \mid \boldsymbol{y})}{q(\boldsymbol{V})} \right\} d\boldsymbol{V}. \tag{7.4}$$

Usually, one is interested in evaluating $p(\boldsymbol{V} \mid \boldsymbol{y})$. However, this may be computationally expensive to find. One can try to approximate $p(\boldsymbol{V} \mid \boldsymbol{y})$ by using $q(\boldsymbol{V})$. Note that, in the following analysis, $q(\boldsymbol{V})$ is not constrained to belong to a particular family of distributions.

Minimising the KL divergence will minimise the difference between the two PDFs. Therefore, summarising, the aim is to find $q(\boldsymbol{V}) \approx p(\boldsymbol{V} \mid \boldsymbol{y})$, if $q(\boldsymbol{V}) = p(\boldsymbol{V} \mid \boldsymbol{y})$ then $\mathrm{KL}(q \,||\, p) = 0$, therefore the lower bound $\mathcal{L}(q)$ has been maximised, which leads to $\log(p(\boldsymbol{y})) = \mathcal{L}(q)$. To locate the optimum approximation $q(\boldsymbol{V})$ one can either, therefore, minimise the $\mathrm{KL}(q \,||\, p)$ or maximise $\mathcal{L}(q)$. The latter is usually adopted as one does not have to evaluate an expectation over p, which in the case present in this study is computationally expensive.

7.2.2 Variational Lower Bound

In the current section, a variational lower bound $\mathcal{L}$ for finding the training data suitable for a sparse GP is introduced, which was originally proposed by Titias [17, 18]. Firstly the full data set is defined as

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \quad \epsilon_n \sim \mathcal{N}(0, \beta^{-1}), \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix} \tag{7.5}$$

where $\boldsymbol{y}$ are the observed outputs, $\boldsymbol{f}$ are the latent training latent functions, $\boldsymbol{x}$ are the training inputs and $\boldsymbol{\epsilon}$ is a noise term. The sparse inputs and outputs are defined as

$$\boldsymbol{u} = \boldsymbol{f}_S(\boldsymbol{z}) + \boldsymbol{\epsilon}, \quad \epsilon_n \sim \mathcal{N}(0, \beta^{-1}), \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_S \end{pmatrix} \tag{7.6}$$

where $\boldsymbol{u}$ are the sparse observations, $\boldsymbol{f}_S$ are the latent function values for the sparse GP drawn from the same latent function as f, $\boldsymbol{z}$ are the sparse inducing inputs, and S are the number of sparse points.

If the true values of the functions are known, the optimum choice of $\boldsymbol{f}_S$ leads to

$$p(\boldsymbol{y} \mid \boldsymbol{f}_S) = p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S). \tag{7.7}$$

Equation (7.7) essentially states that if the optimal $\boldsymbol{f}_S$ is known, knowing $\boldsymbol{f}$ does not alter the probability of witnessing $\boldsymbol{y}$. In the literature [17, 18] the optimal $\boldsymbol{f}_S$ choice is defined as $p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) = p(\boldsymbol{f} \mid \boldsymbol{f}_S)$. Expanding both sides of Bayes' theorem one can show that $p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) = p(\boldsymbol{f} \mid \boldsymbol{f}_S) \Rightarrow p(\boldsymbol{y} \mid \boldsymbol{f}_S) = p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S)$, therefore it does not matter which way it is written.

Having defined the optimal choice of inputs, one can start by defining a prior $p(\boldsymbol{f}, \boldsymbol{f}_S)$ and the likelihood $p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S)$, such that the posterior is

$$p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S) p(\boldsymbol{f}, \boldsymbol{f}_S)}{p(\boldsymbol{y})}. \tag{7.8}$$

Enforcing the condition of the optimal choice of $\boldsymbol{f}_S$, the posterior, from equation (7.8), factorises as:

$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) &= \frac{p(\boldsymbol{y} \mid \boldsymbol{f}_S) p(\boldsymbol{f}, \boldsymbol{f}_S)}{p(\boldsymbol{y})} \\ &= \frac{p(\boldsymbol{y} \mid \boldsymbol{f}_S) p(\boldsymbol{f}_S \mid \boldsymbol{f}) p(\boldsymbol{f})}{p(\boldsymbol{y})} \times \frac{p(\boldsymbol{f}_S)}{p(\boldsymbol{f}_S)} \\ &= \frac{p(\boldsymbol{f}_S \mid \boldsymbol{f}) p(\boldsymbol{f})}{p(\boldsymbol{f}_S)} \times \frac{p(\boldsymbol{y} \mid \boldsymbol{f}_S) p(\boldsymbol{f}_S)}{p(\boldsymbol{y})} \\ &= p(\boldsymbol{f} \mid \boldsymbol{f}_S) p(\boldsymbol{f}_S \mid \boldsymbol{y}) \end{aligned} \tag{7.9}$$

therefore in the optimal case (optimal choice of $\boldsymbol{f}_S$), the posterior PDF factorises as

$$p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) = p(\boldsymbol{f} \mid \boldsymbol{f}_S) p(\boldsymbol{f}_S \mid \boldsymbol{y}). \tag{7.10}$$

To find the difference between two PDFs the KL divergence is required, the KL divergence to find the optimal choice is

$$\mathrm{KL}\left(p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) \,||\, p(\boldsymbol{f} \mid \boldsymbol{f}_S) p(\boldsymbol{f}_S \mid \boldsymbol{y})\right). \tag{7.11}$$

However, the evaluation of $p(\boldsymbol{f}_S \mid \boldsymbol{y})$ has a computational cost of $O(N^3)$, which is shown by the following proof. To start the proof, one first writes Bayes' theorem

$$p(\boldsymbol{f}_S \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{f}_S) p(\boldsymbol{f}_S) \tag{7.12}$$

where $p(\boldsymbol{f}_S)$ is defined as

$$p(\boldsymbol{f}_S) = \mathcal{N}(\boldsymbol{f}_S \mid \boldsymbol{0}, \boldsymbol{K}_{SS}). \tag{7.13}$$

It is known that $p(\boldsymbol{y} \mid \boldsymbol{f}_S)$ is

$$p(\boldsymbol{y} \mid \boldsymbol{f}_S) = \int p(\boldsymbol{y} \mid \boldsymbol{f}) p(\boldsymbol{f} \mid \boldsymbol{f}_S) d\boldsymbol{f} \tag{7.14}$$

where

$$p(\boldsymbol{y} \mid \boldsymbol{f}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{I}\beta^{-1}). \tag{7.15}$$

To find $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$ one defines $\boldsymbol{K}_{NS}$, $\boldsymbol{K}_{SN}$ and $\boldsymbol{K}_{SS}$ as

$$\boldsymbol{K}_{NS} = \begin{pmatrix} k(z_1, x_1) & k(z_2, x_1) & \dots & k(z_S, x_1) \\ k(z_1, x_2) & k(z_2, x_2) & \dots & k(z_S, x_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(z_1, x_N) & k(z_2, x_N) & \dots & k(z_S, x_N) \end{pmatrix}$$

$$\boldsymbol{K}_{SN} = \boldsymbol{K}_{NS}^T$$

$$\boldsymbol{K}_{SS} = \begin{pmatrix} k(z_1, z_1) & k(z_2, z_1) & \dots & k(z_S, z_1) \\ k(z_1, z_2) & k(z_2, z_2) & \dots & k(z_S, z_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(z_1, z_S) & k(z_2, z_S) & \dots & k(z_S, z_S) \end{pmatrix}$$

such that $p(\boldsymbol{f}, \boldsymbol{f}_S)$ is

$$p(\boldsymbol{f}, \boldsymbol{f}_S) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{f}_s \end{pmatrix}; \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}, \begin{bmatrix} \boldsymbol{K}_{NN} & \boldsymbol{K}_{NS} \\ \boldsymbol{K}_{SN} & \boldsymbol{K}_{SS} \end{bmatrix}\right). \tag{7.16}$$

Using the properties of Gaussian distributions $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$ is

$$p(\boldsymbol{f} \mid \boldsymbol{f}_S) = \mathcal{N}(\boldsymbol{f}; \mathrm{E}[\boldsymbol{f} \mid \boldsymbol{f}_S], \mathrm{cov}[\boldsymbol{f} \mid \boldsymbol{f}_S]) \tag{7.17}$$

where

$$\mathrm{cov}[\boldsymbol{f} \mid \boldsymbol{f}_S] = \boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}^T \boldsymbol{K}_{SS}^{-1} \boldsymbol{K}_{NS}$$

and

$$\mathrm{E}[\boldsymbol{f} \mid \boldsymbol{f}_S] = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S.$$

Equation (7.17) can be written more compactly as

$$p(\boldsymbol{f} \mid \boldsymbol{f}_S) = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{K}_{NN} - \boldsymbol{Q}) \tag{7.18}$$

where $\boldsymbol{Q} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^{T}$ and $\boldsymbol{\lambda} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S$.

One now can return to equation (7.14), as $p(\boldsymbol{y} \mid \boldsymbol{f})$ and $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$ have been defined in equations (7.15) and (7.18) respectively. Specifically, $p(\boldsymbol{y} \mid \boldsymbol{f}_S)$ can now be written in the form

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{f}_S) &= \int \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \beta^{-1}\boldsymbol{I})\mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{K}_{NN} - \boldsymbol{Q})d\boldsymbol{f} \\ &= \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \boldsymbol{K}_{NN} - \boldsymbol{Q} + \boldsymbol{I}\beta^{-1}). \end{aligned} \tag{7.19}$$

Now that $p(\boldsymbol{y} \mid \boldsymbol{f}_S)$ (equation (7.19)) and $p(\boldsymbol{f}_S)$ (equation (7.13)) are both defined the $p(\boldsymbol{f}_S \mid \boldsymbol{y})$ in equation (7.12) is therefore

$$p(\boldsymbol{f}_S \mid \boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \boldsymbol{K}_{NN} - \boldsymbol{Q} + \boldsymbol{I}\beta^{-1})\mathcal{N}(\boldsymbol{f}_S \mid 0, \boldsymbol{K}_{SS}). \tag{7.20}$$

The properties of Gaussians and the matrix inverse lemma law can then be used to show that the covariance matrix of $p(\boldsymbol{f}_S \mid \boldsymbol{y})$ is (the full solution is given in Appendix E)

$$\mathrm{cov}(\boldsymbol{f}_S \mid \boldsymbol{y}) = \boldsymbol{K}_{SS} - \boldsymbol{K}_{NS}^{T}\boldsymbol{C}^{-1}\boldsymbol{K}_{NS} \tag{7.21}$$

where $\boldsymbol{C} = \boldsymbol{K}_{NN} + \boldsymbol{I}\beta^{-1}$.

The proof provides evidence that the inversion of $\boldsymbol{K}_{NN}$ is required, with complexity $O(N^3)$, to evaluate $p(\boldsymbol{f}_S \mid \boldsymbol{y})$. It would therefore not be beneficial to locate the sparse points in this

manner, as it would cost the same as creating the GP will all the training data.

Returning to the original problem, where one wants $p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) = p(\boldsymbol{f} \mid \boldsymbol{f}_S)p(\boldsymbol{f}_S \mid \boldsymbol{y})$, which only happens when the optimal choice of sparse points is selected. Using a variational approach, one selects a variational distribution which can be factorised in the same way as equation (7.10). The KL divergence is now written as

$$\mathrm{KL}\left(p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) \mid\mid \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S)\right) \tag{7.22}$$

where $\psi(\boldsymbol{f}_S)$ is an unknown variational distribution. If the KL divergence in equation (7.22) is equal to 0 then is it possible to write

$$p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) = \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S) \tag{7.23}$$

for some distribution $\psi(\boldsymbol{f}_S)$, therefore $p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y})$ has factorised in the same manner as $p(\boldsymbol{f} \mid \boldsymbol{f}_S)p(\boldsymbol{f}_S \mid \boldsymbol{y})$ (equation (7.10)). The KL divergence is written as

$$\mathrm{KL}(p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}) \mid\mid q(\boldsymbol{f}, \boldsymbol{f}_S)) \tag{7.24}$$

where

$$q(\boldsymbol{f}, \boldsymbol{f}_S) = \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S). \tag{7.25}$$

Minimising KL is equivalent to maximising the lower bound

$$\mathcal{L} = \int\int q(\boldsymbol{f}, \boldsymbol{f}_S) \log\left[\frac{p(\boldsymbol{f}, \boldsymbol{f}_S, \boldsymbol{y})}{q(\boldsymbol{f}, \boldsymbol{f}_S)}\right] d\boldsymbol{f} d\boldsymbol{f}_S. \tag{7.26}$$

Enforcing the factorisation property of the variational distribution ($q(\boldsymbol{f}, \boldsymbol{f}_S) = \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S)$)

$$\mathcal{L} = \int\int \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log\left\{\frac{p(\boldsymbol{f}, \boldsymbol{f}_S, \boldsymbol{y})}{\psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S)}\right\} d\boldsymbol{f} d\boldsymbol{f}_S. \tag{7.27}$$

It is known that $p(\boldsymbol{f}, \boldsymbol{f}_S, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{f})p(\boldsymbol{f} \mid \boldsymbol{f}_S)p(\boldsymbol{f}_S)$, so the lower bound can be written as

$$\begin{aligned}
\mathcal{L} &= \int\int \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log\left\{\frac{p(\boldsymbol{y} \mid \boldsymbol{f})p(\boldsymbol{f} \mid \boldsymbol{f}_S)p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S)}\right\} d\boldsymbol{f}d\boldsymbol{f}_S \\
&= \int\int \psi(\boldsymbol{f}_S)p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log\left\{\frac{p(\boldsymbol{y} \mid \boldsymbol{f})p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right\} d\boldsymbol{f}d\boldsymbol{f}_S \\
&= \int \left\{ \psi(\boldsymbol{f}_S) \underbrace{\left[\int p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log p(\boldsymbol{y} \mid \boldsymbol{f}) d\boldsymbol{f}\right]}_{\log G(\boldsymbol{f}_S, \boldsymbol{y})} + \log\left(\frac{p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right) \right\} d\boldsymbol{f}_S
\end{aligned} \tag{7.28}$$

To evaluate $\log G(\boldsymbol{f}_S, \boldsymbol{y})$, one first needs an expression for $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$, which is shown in equation (7.18) to be

$$p(\boldsymbol{f} \mid \boldsymbol{f}_S) = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{K}_{NN} - \boldsymbol{Q}) \tag{7.29}$$

where $\boldsymbol{Q} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T$ and $\lambda = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S$. Defining for simplicity

$$\mathrm{Cov}\left[\boldsymbol{f} \mid \boldsymbol{f}_S\right] = \boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{SN} = \boldsymbol{\Sigma}$$

allows us to write

$$p(\boldsymbol{f} \mid \boldsymbol{f}_S) = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{\Sigma}). \tag{7.30}$$

To evaluate $\log G$ in equation (7.28)

$$\log G(\boldsymbol{f}_S, \boldsymbol{y}) = \int \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{\Sigma}) \log p(\boldsymbol{y} \mid \boldsymbol{f}) d\boldsymbol{f} \tag{7.31}$$

it is known that

$$p(\boldsymbol{y} \mid \boldsymbol{f}) \propto \frac{1}{\left(2\pi^{\frac{N}{2}}\right)} \mid \beta^{-1}\boldsymbol{I} \mid^{\frac{1}{2}} \exp\left(-\frac{\beta}{2}(\boldsymbol{y} - \boldsymbol{f})^T(\boldsymbol{y} - \boldsymbol{f})\right) \tag{7.32}$$

therefore

$$\begin{aligned}\log p(\boldsymbol{y} \mid \boldsymbol{f}) &= -\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}(\boldsymbol{y}-\boldsymbol{f})^T(\boldsymbol{y}-\boldsymbol{f}) \\ &= -\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}\left[\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{f} + \boldsymbol{f}^T\boldsymbol{f}\right]\end{aligned} \tag{7.33}$$

which allows equation (7.31) to be written as

$$\begin{aligned}\log G(\boldsymbol{f}_S, \boldsymbol{y}) &= \int p(\boldsymbol{f} \mid \boldsymbol{f}_S)\left(-\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}[\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{f} + \boldsymbol{f}^T\boldsymbol{f}]\right)d\boldsymbol{f} \\ &= -\frac{N}{2}\log(2\pi\beta^{-1})\int p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f} - \frac{\beta}{2}\int [\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{f} + \boldsymbol{f}^T\boldsymbol{f}]p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f}. \\ &= -\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}\underbrace{\int [\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{f} + \boldsymbol{f}^T\boldsymbol{f}]p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f}}_{\text{Int}}.\end{aligned} \tag{7.34}$$

Expanding the integral, Int (in the previous equation), it can be shown that

$$\text{Int} = \underbrace{\boldsymbol{y}^T\boldsymbol{y}\int p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f}}_{\text{Part 1}} - \underbrace{2\boldsymbol{y}^T\int \boldsymbol{f}p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f}}_{Part2} + \underbrace{\int \boldsymbol{f}^T\boldsymbol{f}p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f}}_{Part3}. \tag{7.35}$$

The result of solving the integral of Part 1 using knowledge that $\int p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f} = 1$

$$\text{Part 1} = \boldsymbol{y}^T\boldsymbol{y}\int p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f} = \boldsymbol{y}^T\boldsymbol{y} = \sum_{i=1}^{N} y_i^2.$$

Using the proof shown in Appendix F.1, Part 2 can be shown to be

$$\text{Part 2} = 2\boldsymbol{y}^T\int \boldsymbol{f}p(\boldsymbol{f} \mid \boldsymbol{f}_S)d\boldsymbol{f} = -2\boldsymbol{y}^T\boldsymbol{\lambda} = \sum_{i=1}^{N} -2y_i\lambda_i$$

Using equation (7.30) and the proof in Appendix F.2 one can write part 3 as

$$\text{Part 3} = \int \boldsymbol{f}\boldsymbol{f}^T \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \int f_i^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i = \sum_{i=1}^{N} \Sigma_{ii} + \lambda_i^2.$$

The final integral, Int, is therefore

$$\begin{aligned} \text{Int} &= \sum_{i=1}^{N} y_i^2 - 2y_i\lambda_i + \lambda_i^2 + \Sigma_{ii} \\ &= \sum_{i=1}^{N} (y_i - \lambda_i)^2 + \Sigma_{ii} \end{aligned} .$$

Using the solution for the integral, Int, $\log G(\boldsymbol{f}_S, \boldsymbol{y})$ can be written as

$$\begin{aligned} \log G(\boldsymbol{f}_S, \boldsymbol{y}) &= -\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}\sum_{i=1}^{N}[(y_i - \lambda_i)^2 + \Sigma_{ii}] \\ &= -\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}\sum_{i=1}^{N}(y_i - \lambda_i)^2 - \frac{\beta}{2}\text{Tr}(\boldsymbol{\Sigma}) \end{aligned} . \tag{7.36}$$

therefore, $\log G(\boldsymbol{f}_S, \boldsymbol{y})$ is

$$\log G(\boldsymbol{f}_S, \boldsymbol{y}) = \log(\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \beta^{-1}\boldsymbol{I})) - \frac{\beta}{2}\text{Tr}(\boldsymbol{\Sigma}) \tag{7.37}$$

where

$$\log\left(\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\lambda}, \beta^{-1}\boldsymbol{I}\right)\right) = -\frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}\sum_{i=1}^{N}[(y_i - \lambda_i)^2].$$

Referring to equation (7.28) the lower bound can, therefore, be written as

$$\mathcal{L} = \int \psi(\boldsymbol{f}_S)\left[\log G(\boldsymbol{f}_S, \boldsymbol{y}) + \log\left(\frac{p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right)\right] d\boldsymbol{f}_S \tag{7.38}$$

where $\log G(\boldsymbol{f}_S, \boldsymbol{y}) = \log(\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \beta^{-1}\boldsymbol{I})) - \frac{\beta}{2}\text{Tr}(\boldsymbol{\Sigma})$.

The lower bound can be written as a functional of $\boldsymbol{z}$ and ψ

$$\begin{aligned}\mathcal{L}(\boldsymbol{z},\psi) &= \int \psi(\boldsymbol{f}_S)\left[\log G(\boldsymbol{f}_S,\boldsymbol{y}) + \log\left(\frac{p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right)\right] d\boldsymbol{f}_S \\ &= \int \psi(\boldsymbol{f}_S)\log\left(\frac{G(\boldsymbol{f}_S,\boldsymbol{y}_S)p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right) d\boldsymbol{f}_S.\end{aligned} \tag{7.39}$$

The following method allows for the maximisation of the lower bound on $\mathcal{L}$ without having to find the optimal distribution ψ^*. This is advantageous because, as stated in Section 7.2.2, there are problems associated with finding ϕ^*. It is much simpler for the Jensen's inequality to be used to evaluate the lower bound on $\mathcal{L}$

$$\mathcal{L} \geq \mathcal{L}_{LB} = \log \int \left[\psi(\boldsymbol{f}_S)\frac{G(\boldsymbol{f}_S,\boldsymbol{y})p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right] d\boldsymbol{f}_S \tag{7.40}$$

therefore the lower bound $\mathcal{L}_{LB}$ can be written as

$$\begin{aligned}\mathcal{L}_{LB} &= \log\left[\int \psi(\boldsymbol{f})\left(\frac{G(\boldsymbol{f}_S,\boldsymbol{y})p(\boldsymbol{f}_S)}{\psi(\boldsymbol{f}_S)}\right) d\boldsymbol{f}_S\right] \\ &= \log \int G(\boldsymbol{f}_S,\boldsymbol{y})p(\boldsymbol{f}_S)d\boldsymbol{f}_S.\end{aligned} \tag{7.41}$$

Recalling that $G(\boldsymbol{f}_S,\boldsymbol{y})$ is defined as

$$G(\boldsymbol{f}_S,\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \beta^{-1}\boldsymbol{I}) - \exp\left(\frac{\beta}{2}\mathrm{Tr}(\boldsymbol{\Sigma})\right)$$

one can write the $\mathcal{L}_{LB}$ as

$$\mathcal{L}_{LB} = \log \int \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \beta^{-1}\boldsymbol{I})p(\boldsymbol{f}_S)d\boldsymbol{f}_S - \exp\left(\frac{\beta}{2}\mathrm{Tr}(\boldsymbol{\Sigma})\right)\int p(\boldsymbol{f}_S)d\boldsymbol{f}_S. \tag{7.42}$$

Using the knowledge that $\int p(\boldsymbol{f}_S)d\boldsymbol{f}_S = 1$ and that $\boldsymbol{\lambda} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S$ one can write

$$\mathcal{L}_{LB} = \log \int \underbrace{\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S, \beta^{-1}\boldsymbol{I})\mathcal{N}(\boldsymbol{f}_S \mid \boldsymbol{0}, \boldsymbol{K}_{SS})d\boldsymbol{f}_S}_{\text{Int2}} - \exp\left(\frac{\beta}{2}\mathrm{Tr}(\boldsymbol{\Sigma})\right) \tag{7.43}$$

The integral (Int2) is evaluated using standard expressions for Gaussian distributions, shown in Appendix G, such that the objective function becomes

$$\mathcal{L}_{LB} = \log\left[\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{0}, \beta^{-1}\boldsymbol{I} + \boldsymbol{Q})\right] - \frac{\beta}{2}\mathrm{Tr}(\boldsymbol{\Sigma}). \tag{7.44}$$

$\mathcal{L}_{LB}$ can be used as a cost function for finding $\boldsymbol{z}$ and the sparse GP hyperparameters. Expanding using the properties of multivariate Gaussians, the cost function can be written as

$$\mathcal{L}_{LB} = -\frac{1}{2}\left(N\log(2\pi) + \log\left(\mid \boldsymbol{I}\beta^{-1} + \boldsymbol{Q} \mid\right) + \boldsymbol{x}^T\boldsymbol{x}(\boldsymbol{I}\beta^{-1} + \boldsymbol{Q})^{-1} - \beta\mathrm{Tr}(\boldsymbol{\Sigma})\right) \tag{7.45}$$

where

$$\boldsymbol{Q} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T \qquad \boldsymbol{\Sigma} = \boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{SN}$$

$$\boldsymbol{K}_{NS}(i,j) = k(\boldsymbol{x}_i, \boldsymbol{x}_{m,j}), \qquad \boldsymbol{K}_{SS}(i,j) = k(\boldsymbol{x}_{m,i}, \boldsymbol{x}_{m,j})$$

The cost complexity of the method used to select the sparse points is, therefore, $O(NM^2)$.

7.2.3 Optimisation of the Sparse Points

Greedy Selection / Optimisation

To locate the optimal sparse points requires a combinatorial search, which is expensive. In the current thesis, a greedy selection algorithm is used. The greedy selection will aim to pick the best sparse points according to the selection criterion, which in the case of the lower bound is $\mathcal{L}_{LB}$. Before the greedy selection algorithm is run, an initial sparse set (Z) is chosen from clusters identified by the K-means algorithm. Such an approach ensures that the initial points are chosen from clusters that already exist in the data, providing a reasonable initial estimate for training points. An initial choice of hyperparameters are also selected. The greedy algorithm runs in the following manner

1. Start with the initial sparse set, chosen from each cluster of the K-means clustering solution. The remaining set is, therefore, $(N - Z)$; the remaining set contains training inputs $\boldsymbol{x}$.

2. Create a randomly chosen working set W which consists of C candidate training inputs from the remaining set, where is a C is a user-defined number of candidate inputs.

3. $\mathcal{L}_{LB}$ is calculated for the working set.

4. The maximum value of $\mathcal{L}_{LB}$ from values in the working set is located, and the corresponding training point is added into the sparse set and removed from the remaining set.

5. The hyperparameters corresponding to the sparse set are optimised to maximise the lower bound and updated.

6. Repeat steps 2 to 5 until $\mathcal{L}_{LB}$ has converged.

According to Titsias [17, 18], this is viewed as an EM-like algorithm, where at the E step, a point is added to the sparse set, and at step M one updates the hyperparameters. The value of $\mathcal{L}_{LB}$ increases for both the E and M steps; this makes it easier to judge convergence. Consequently, adding additional sparse points will eventually maximise the lower bound and minimise the KL divergence $\mathrm{KL}(q(\boldsymbol{f}, \boldsymbol{f}_S) \mid\mid p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y}))$. As the sparse points are selected randomly at each iteration, it is impossible to guarantee that the final selection of sparse points is optimal.

Figure 7.3 displays the results of the variational sparse GP method applied to the same data used for Figures 7.1 and 7.2 (a sine wave without any measure noise). It is interesting to note that the sparse points are selected at the crest and trough of the sine wave.

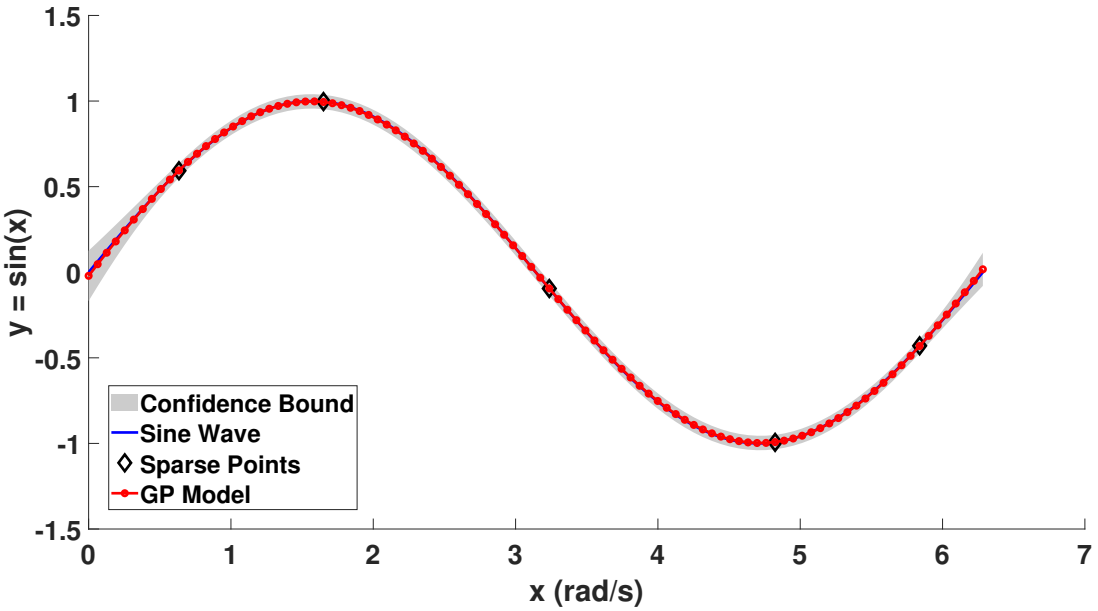


Figure 7.3: Sparse GP with the training points selected using the variational sparse GP method.

7.3 Summary

In the current chapter, the variational sparse GP method by Titsias [17, 18] is presented in detail. Firstly a brief overview of the variational Bayes framework is given, which is followed by the methodology of the variational sparse GP and derivation of the lower bound used to locate the optimal sparse points. Finally, the chapter is finished with a description of the optimisation method used to locate the optimal sparse points.

The variational sparse GP algorithm that was described in the chapter was firstly introduced by Titias [17, 18], the detailed methodology of the approach is shown here, to create a self-contained document. The code implemented by the author used the algorithms shown in Section 7.2.2. The algorithms which include sparse GP implementations and optimisation techniques were written and executed in Matlab.

It is worth noting that Hensman et al. [68] introduced a different lower bound for the likelihood which is found by using Jensen's inequality and the Kullbeck-Lieber (KL) divergence, which allowed for a Stochastic Variational Inference approach to be used. The method proposed by Hensman et al. [68] and Titsias [17, 18] both attempt the minimise the difference between the sparse GP and the full GP. The approach proposed by Hensman et al. is not required

here because the datasets are not large enough to warrant such an approach. A detailed description of the approach described in [68] is shown in Appendix D. Titsias [17, 18] states that the variational sparse GP method is closely related to the methods described in [69, 70, 71] where the methods also use a variational approach but only for the estimation of the kernel hyperparameters.

Chapter 8

Variational Sparse Gaussian Process Applied to Bo105 On-axis Responses

In the previous chapter, the variational sparse GP approach was introduced. The variational sparse GP has the potential to reduce the size of the training set required to make accurate predictions. The variational sparse GP reduces the time complexity of the training phase from $O(N^3)$ to $O(NM^2)$, where in the current example, N is the number of data points in the full dataset, M is the number of training points and $M < N$. The reduction in the time complexity of training demonstrates the scalability of the approach, as more data would likely be required than shown in the thesis to train the GP model for a full flight test campaign. For the prediction phase of the GP model the complexity is $O(N)$ for the mean and $O(N^2)$ for the variance. For the sparse GP model the time complexity for prediction of the mean is $O(M)$ and for the variance $O(M^2)$; therefore, it is less of a concern compared to the cost associated with the training phase. In the current chapter, the variational sparse approach is used to create sparse GP models to predict the response of the pitch rate, roll rate, yaw rate and heave. For the sake of brevity in the current chapter, the variational sparse GP will be referred to simply as the 'sparse GP'. As shown in Chapter 5, the ARX input structure containing all of the current pilot inputs with two lagged response term produces more accurate response and acceleration predictions for the pitch, roll and yaw. In the case of the heave, the ARX input structure containing all of the current pilot inputs with one lagged heave term produced the

most accurate response and acceleration predictions. In Chapter 6, different hyperparameter structures were investigated, in which for all four responses (pitch rate, roll rate, yaw rate and heave) more accurate predictions were produced using one hyperparameter than using a hyperparameter assigned to each input of the ARX structure. The variational sparse GP aims to produce similar predictions to that of the 'full' GP (the GP shown in Chapters 5 and 6 using 32 training points is referred to as the full GP throughout the rest of the thesis). For the sparse GP results in the current chapter, it was decided to use the ARX input structure containing all of the current pilot inputs with two lagged response terms utilising one hyperparameter for the pitch rate, roll rate and yaw rate as this approach produce the most accurate results in Chapters 5 and 6. For the heave response, the ARX input structure contains all of the current pilot inputs with one lagged heave term utilising one hyperparameter.

Firstly, the sparse GP models were trained on 32 training points, to investigate if the sparse GP and greedy search approach can create GP models for the pitch rate, roll rate, yaw rate, and heave that produce acceptable predictions that are comparable to that of the full GP. The sparse GP models are compared to the real flight test data, the corresponding FLIGHTLAB model and the full GP model. These results are followed by sparse GP models trained on a reduced training, to assess if the sparse GP model can use fewer points than the full GP and produced a comparable model. Regarding scalability to much larger data sets, if the number of training points could be reduced, this would benefit the training and prediction time of the GP model. The reduction in training points would allow the GP to be used to analyse more extensive datasets, with more manoeuvres, in the future.

As with the previous Chapters 5 and 6, for the sake of brevity, it is not beneficial to show the OSAP, as the FMP are the more stringent test of the model's fidelity. However, the OSAP results can be seen in Appendix H.

For a recap, the cases used in the current chapter are given in Table 8.1.

Manoeuvre	Prediction	Input structure	Case
3-2-1-1 Longitudinal input	Pitch rate	All of the current pilot inputs with two lagged pitch rate terms	1
3-2-1-1 Lateral input	Roll rate	All of the current pilot inputs with two lagged roll rate terms	2
3-2-1-1 Pedal input	Yaw rate	All of the current pilot inputs with two lagged yaw rate terms	3
3-2-1-1 Collective input	Heave	All of the current pilot inputs with one lagged heave term	4

Table 8.1: GP configurations for the investigation of the use of Sparse GPs

8.1 Sparse Gaussian Process - 32 Points

In the current section, the sparse GP is shown using 32 training points, firstly to verify the code and to test if the greedy search method can produce results that are comparable to that of the full GP. The sparse GP code was created by the author employing the methodology described in Chapter 7.

To begin training a variational sparse GP model, one must choose a small number of initial training data points. It would be beneficial to select the initial training points that provide information for different regions of the dataset so that sparse GP approach has some knowledge of the dynamics of the response. Therefore, clustering the GP input ($\boldsymbol{x}$) allows a training point to be selected from each cluster and ensures that the initial training points that are chosen are from different regions of the dataset. Here, the K-means clustering algorithm was used to locate the clusters where an initial training point from each cluster was selected. Such an approach ensures that the initial points are chosen from clusters that already exist in the data, providing a reasonable initial estimate for training points. For the sparse results presented in the current chapter, the K-means clustering algorithm [26] was used to identify five clusters (the author selected this and the number of clusters is user-defined for the K-means algorithm), allowing the variational sparse GP to be initiated using five points (one from each cluster).

8.1.1 Pitch Rate

As previously mentioned, the OSAP figures are not shown in the main text. However, figures can be seen in Appendix H. The lower bound and the hyperparameter convergence can be seen in Figures 8.1 and 8.2, respectively, for case 1, Table 8.1. The convergence is shown after the initial 5 training points were chosen using the K-means clustering algorithm. The hyperparameters converge at 9 training points. However, the lower bound has not converged after 9 training points; therefore, more training points are required for the sparse GP lower bound to convergence. The convergence of the lower bound in Figure 8.1 was judged to have converged after 12 training points even though up to 32 were investigated. Note, for the judgment of convergence, no criteria was used it was based authors opinion. The use of a reduced training set is conducted in the next Section 8.2.1; where 12 training points are used to create the sparse GP model. In the current section, 32 training points are used to create the GP models.

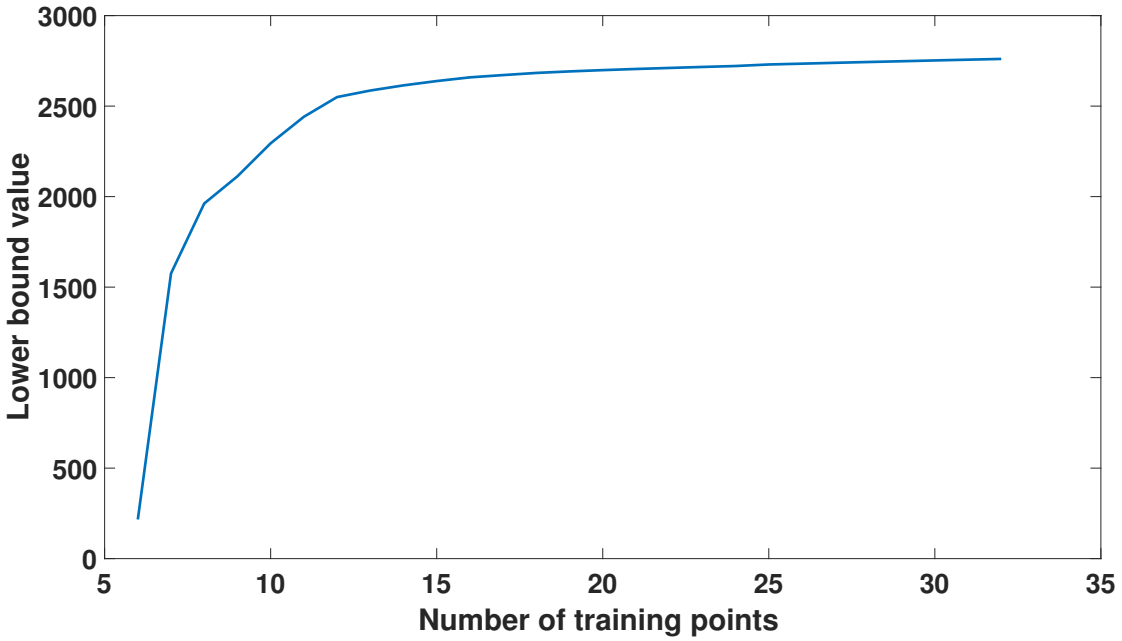


Figure 8.1: The convergence of the lower bound for the pitch rate sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 1, Table 8.1).

Figure 8.3 displays the rotorcraft inputs used for the pitch rate sparse GP. All of the pilot inputs shown in Figure 8.3 are used in the input structure for the pitch rate sparse GP. The training points can be seen in Figure 8.3, which were identified by the sparse GP approach. It

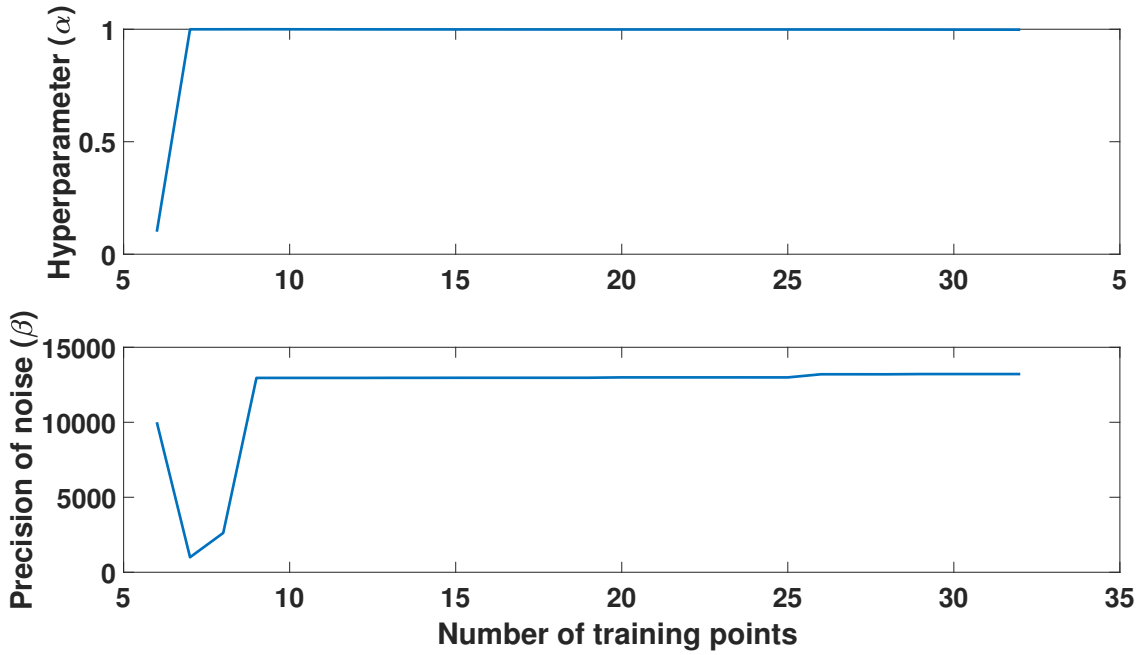
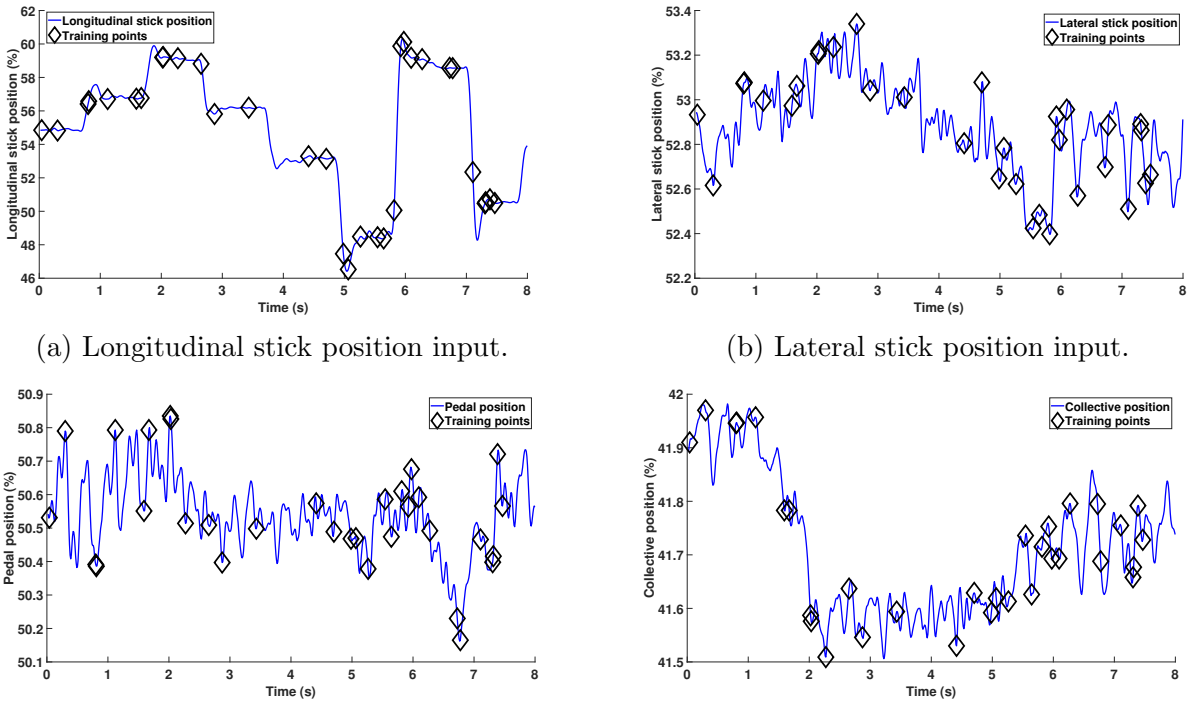


Figure 8.2: The convergence of the hyperparameters for the pitch rate sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 1, Table 8.1).



(a) Longitudinal stick position input.

(b) Lateral stick position input.

(c) Pedal position input.

(d) Collective position input.

Figure 8.3: Bo105 rotorcraft inputs for the 3-2-1-1 longitudinal cyclic input with 32 training points selected by the sparse GP approach.

is interesting to see that the training points are spread across the whole time series and that there are training points near the highest and smallest longitudinal value.

Figure 8.4 displays two sets of FMP GP realisations. The first set of realisations is the sparse GP FMP with 32 training points using in input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising one hyperparameter (case 1, Table 8.1) displayed by red realisations. It can be seen that the realisations capture the dynamics of the pitch rate response. The second set of FMP GP realisations is the full GP using 32 training points, displayed as green realisations. Both sets of full model predictions were realised using 10,000 Monte Carlo runs. The grey shaded areas occur when the two GP predictions (sparse GP and the full GP) overlap.

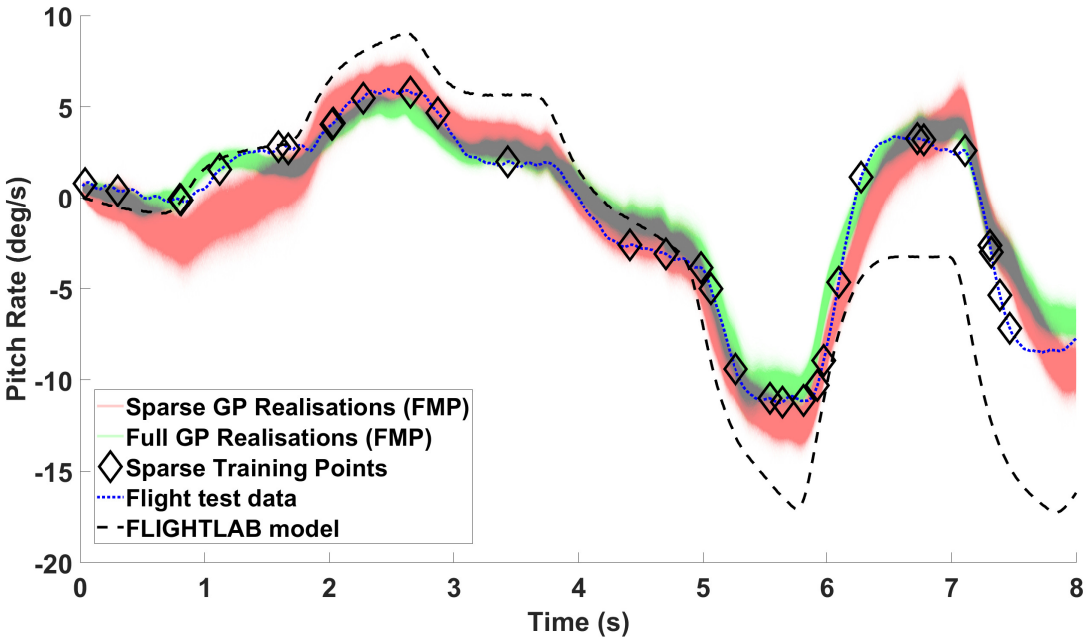


Figure 8.4: Realisations of sparse GP full model pitch rate predictions using 32 training points utilising one hyperparameter (α) (case 1, Table 8.1), with comparison to the corresponding full GP.

The FLIGHTLAB model and flight test data are shown in Figure 8.4 for visual comparison. Quantifying the difference, the similarity measure between the mean FMP sparse GP model and the flight test data is 8.54, compared to the full GP 5.52. The two models are quite similar; the difference between them could be due to the random element of the greedy approach for selecting the training points or the optimisation of the hyperparameters. The optimisation of the hyperparameters uses the MATLAB function fmincon to maximise the lower bound and to locate the optimal hyperparameter; this differs from the optimisation of the full GP, which uses simulated annealing to select samples from the posterior hyperparameter distribution.

The difference in the optimisation could have an impact on the hyperparameter selection. For comparison sake the full GP had hyperparameters, $\alpha = 0.992$ and $\beta = 5.946 \times 10^5$ and the hyperparameters for the sparse GP are $\alpha = 0.998$ and $\beta = 1.322 \times 10^4$. The difference is notable for the precision of noise (β), and this could have affected the FMP realisations. The maximisation of the lower bound has a reduced computation cost compared to taking samples from the posterior hyperparameter distribution (see section 2.1.3). Therefore for large datasets, such as a full flight test campaign, it would be beneficial to use the lower bound to locate the hyperparameters.

Comparing the GP FMP realisations for the sparse GP and the full GP, the sparse GP has a greater range. The wider bounds display reduced confidence; this could be due to what has been previously discussed, i.e., that the greedy approach has a random element to selecting the training points or the optimisation of the hyperparameters. However, the sparse GP and the full GP does capture the dynamics of the flight test data. Both GP models still outperform the FLIGHTLAB model, which has a similarity measure of 75.02.

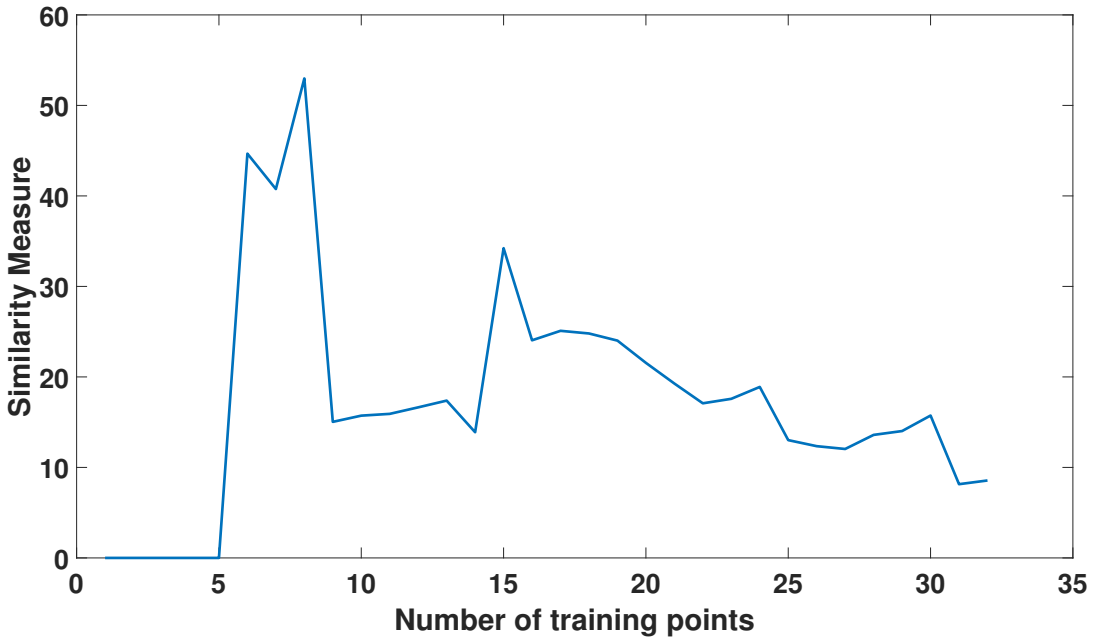


Figure 8.5: The similarity measure between the pitch rate flight test data and the mean of FMP realisations for a range of training points.

The challenges associated with the FMP predictions when trained on the OSAP are highlighted by Girard [46]. It is also worth stating that the full model predictions are very sensitive to which

inputs are selected. The sensitivity of the FMP predictions can be seen in Figure 8.5, which displays the similarity measure between the mean GP FMP realisations and the flight test data for 5 to 32 training points. Adding one more training point can reduce the performance of the model in terms of similarity between the mean FMP realisations and the flight test data, which is apparent when examining Figure 8.5 where it can be seen that the similarity measure for 14 training points is less than for 15 training points. The result could be due to the random element of the greedy selection missing the next optimal point. However, the lower bound is attempting to optimise for one step ahead predictions and not the full model predictions. The peak (between 5 and 9 training points) in Figure 8.5 is likely to be due to the sparse GP hyperparameters and lower bound having not fully converged, and the full model predictions are highly inaccurate until the convergence has occurred. Note that the monotonic increase in the lower bound may not be the case for the similarity measure using the mean FMP realisations.

8.1.2 Roll Rate

The roll rate sparse GP convergence of the lower bound and hyperparameters are shown in Figures 8.6 and 8.7, respectively, for case 2, Table 8.1. The hyperparameters have converged after 10 training points. The lower bound is judged to have converged after 20 training points, therefore in the next Section 8.2.2 (where a reduced training set is investigated), the roll rate response sparse GP will use only 20 training points. In the current section, 32 training points are used to investigate if the greedy approach can produce results that are comparable to the roll rate full GP.

Figure 8.8 displays the inputs used for the roll rate sparse GP input structure. The diamonds in Figure 8.8 indicate the training points, which are automatically selected during the sparse GP approach. As with the pitch rate training points, the roll rate training points are spread across the time series. The selection of the training points provides evidence that informative points are spread along the time series and are not concentrated in a particular part of the manoeuvre.

Two sets of roll rate GP FMP realisations are displayed in Figure 8.9. The first set of GP FMP realisations are for the sparse GP model using 32 training points and an input/hyperparameter

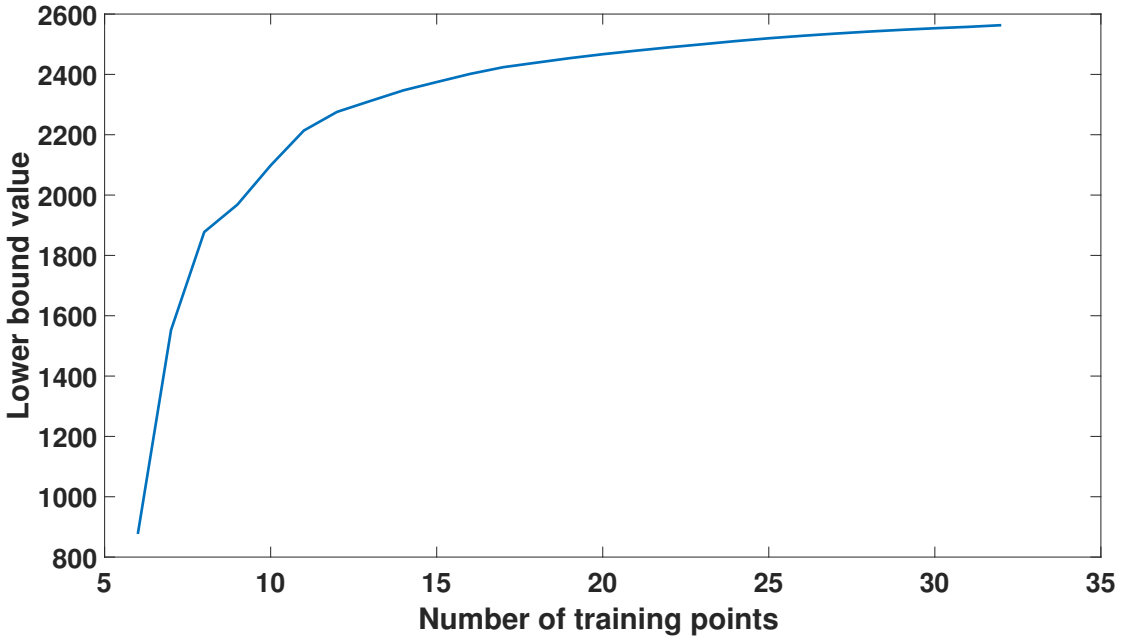


Figure 8.6: The convergence of the lower bound for the roll rate sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 2, Table 8.1).

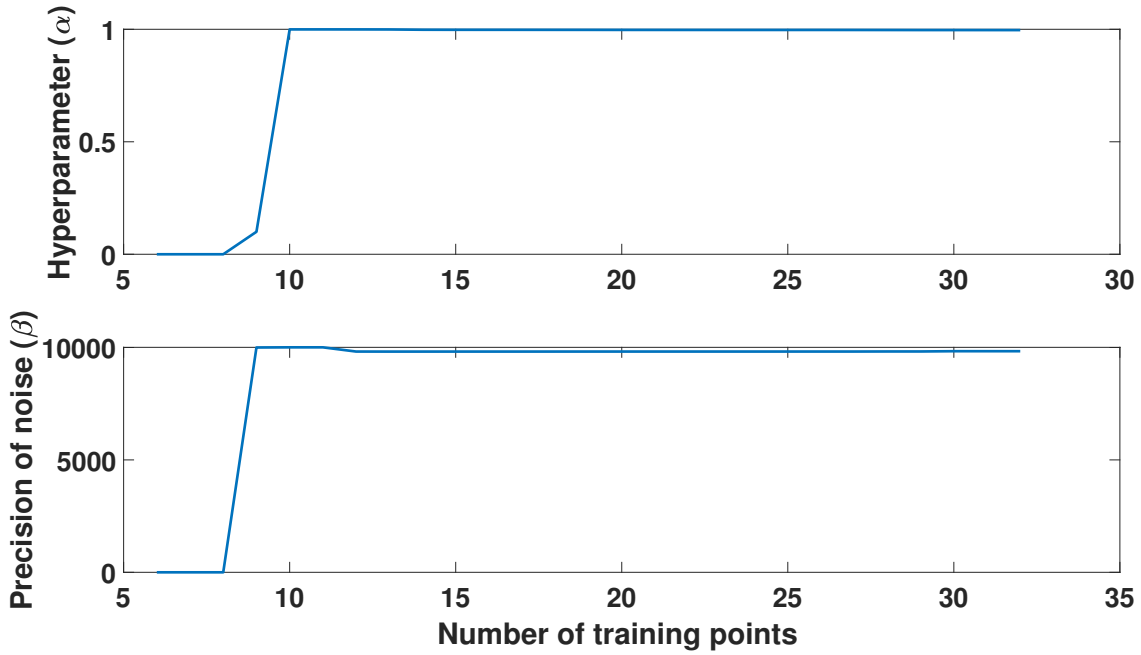
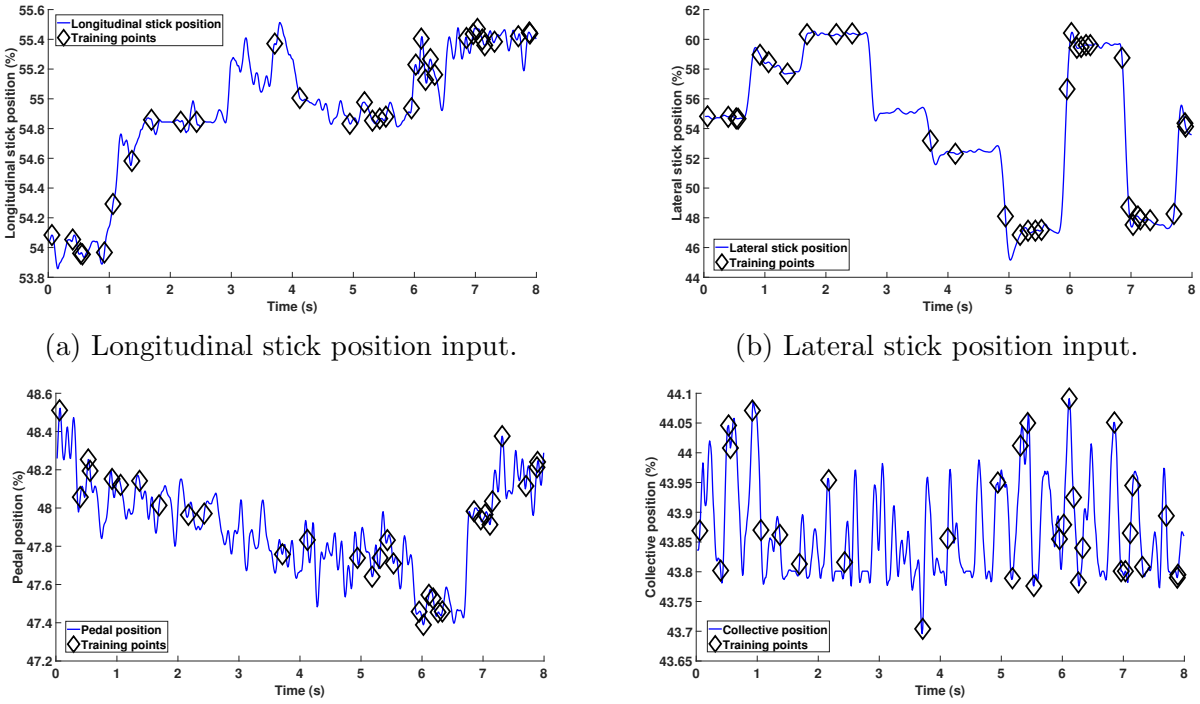


Figure 8.7: The convergence of the hyperparameters for the roll rate sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 2, Table 8.1).



(a) Longitudinal stick position input.

(b) Lateral stick position input.

(c) Pedal position input.

(d) Collective position input.

Figure 8.8: Bo105 rotorcraft inputs for the 3-2-1-1 lateral cyclic input with 32 training points selected by the sparse GP approach.

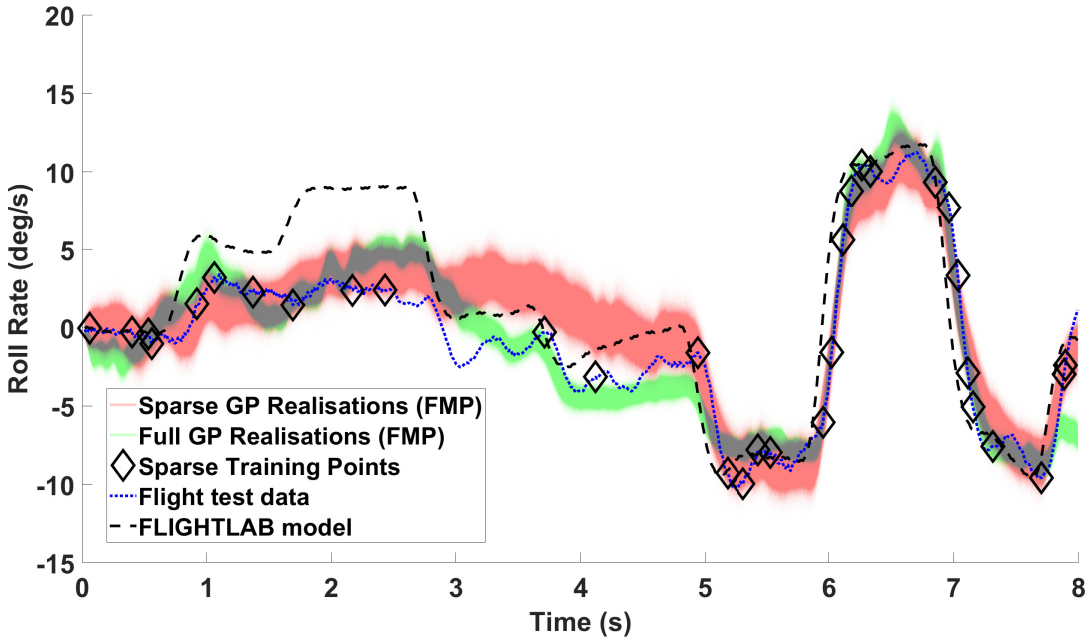


Figure 8.9: Realisations of sparse GP full model roll rate predictions using 32 training points utilising one hyperparameter (α) (case 2, Table 8.1), with comparison to the corresponding full GP.

structure containing all of the current pilot inputs with two lagged roll rate terms utilising one hyperparameter (case 2, Table 8.1) are displayed in Figure 8.9 as red realisations. For comparison, the full GP FMP realisations are shown in Figure 8.9 by green realisations. As with the pitch rate response shown in Figure 8.4, the roll rate predictions in Figure 8.9 captured the dynamics of the response and produced a comparable model to the full GP. The results from the FMP provides evidence that the sparse GP can closely replicate what the full GP predicts for this manoeuvre. The similarity measure between the sparse GP mean FMP realisations to the flight test data is 15.29, compared to the full GP, 10.83, and the FLIGHTLAB model 42.43. It is clear that both GP models outperform the FLIGHTLAB model. The sparse GP does produce a slightly worse model in terms of similarity measure compared to the full GP, which could be due to what has been previously discussed that the greedy approach may not have selected the optimal training points.

8.1.3 Yaw Rate

Figures 8.10 and 8.11 show the convergence of the lower bound and hyperparameters, respectively, for case 3, Table 8.1. The convergence of the hyperparameters α and β happen after 10 training points. However, the lower bound had not converged, the lower bound appears to converge after approximately 15 points. In the current section, 32 training points are used for the sparse GP models.

The inputs used for the input structure for the yaw rate sparse GP are shown in Figure 8.12. The black diamonds indicate the training points, which are automatically selected by the sparse GP approach. It can be seen that in Figure 8.12c that the black diamonds are concentrated around the highest and lowest values of the pedal position. The sparse GP attempts to select the more 'information-rich' (i.e. those that lead to the greatest increase in the lower bound) training points. The training points shown in Figure 8.12c are selected around the highest and lowest values; therefore, the selected points must have more 'information'. The selection of points seems reasonable as at highest and lowest pedal position the largest displacement from the equilibrium is presented, in which should have the largest effect on the yaw rate, and therefore provide more information about the dynamics of the response.

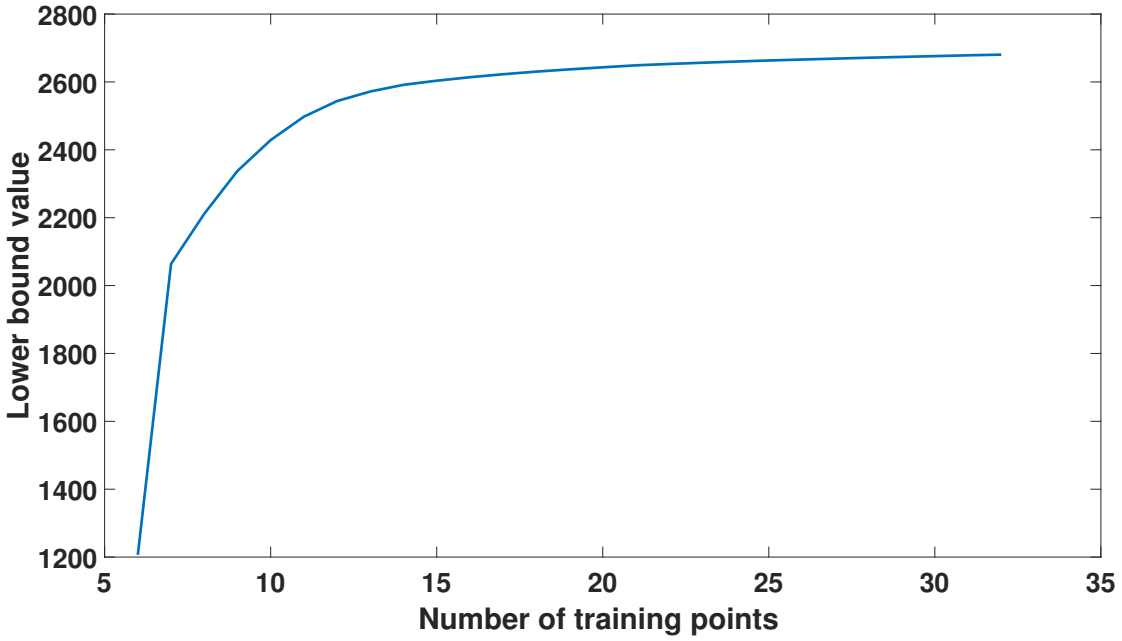


Figure 8.10: The convergence of the lower bound for the yaw rate sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 3, Table 8.1).

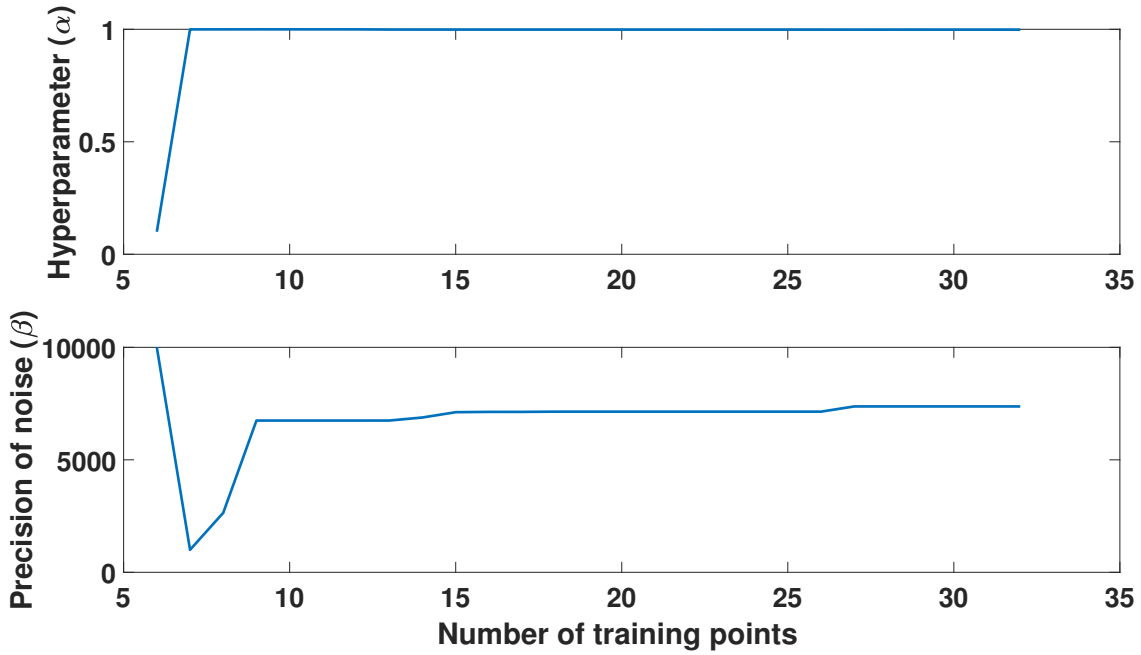
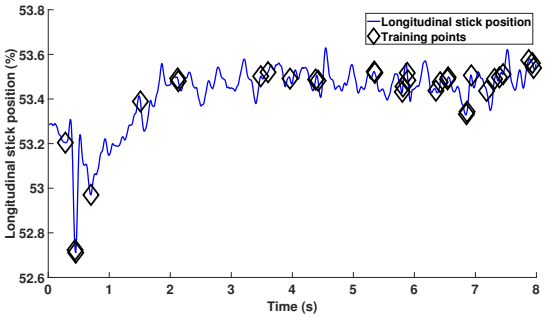
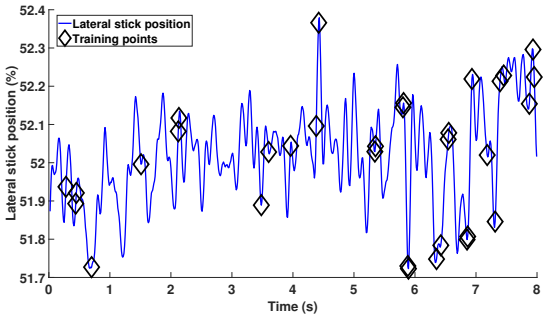


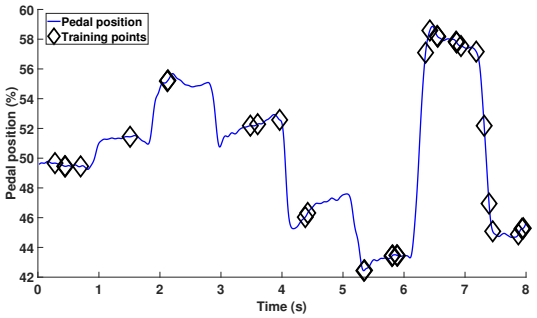
Figure 8.11: The convergence of the hyperparameters for the yaw rate sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 3, Table 8.1).



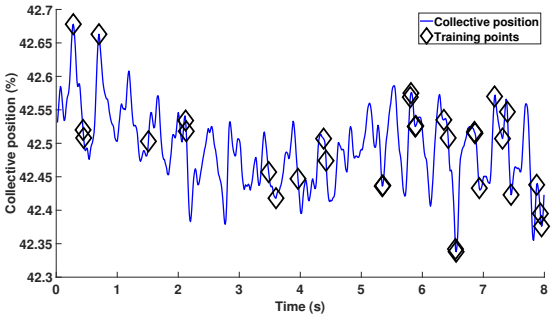
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 8.12: Bo105 rotorcraft inputs for the 3-2-1-1 pedal input with 32 training points selected by the sparse GP approach.

Figure 8.13 displays two sets of GP FMP realisations. The first ensemble of FMP realisations are for the sparse GP using 32 training points and an input/hyperparameter structure containing all of the current pilot inputs with two lagged yaw rate terms utilising one hyperparameter (case 3, Table 8.1), shown as red realisations. The second set of GP FMP realisations are for the full GP, which are displayed by green realisations. Visually, the sparse GP FMP realisations (red realisations) are slightly worse than the full GP model (green realisations) across the time series. The grey regions in Figure 8.13 show the shared realisations. The two GP models in Figure 8.13 produce better predictions than that of the FLIGHTLAB model, especially towards the end of the time series. Quantifying the difference between predictions, in terms of similarity measure between the sparse GP mean FMP realisations and the flight test data is 129.45, compared to the full GP similarity measure which is 53.26, and the similarity measure for the corresponding FLIGHTLAB model is 370.08. The full GP creates more accurate predictions than that of the sparse GP; this could be due to the random selection in the greedy optimisation. Another reason, which has been previously discussed, is that the sparse GP is attempting to minimise the lower bound based on OSAP and not FMP. If one examines the OSAP, they are very

similar; while the FMP predictions are quite sensitive to changes in the training points.

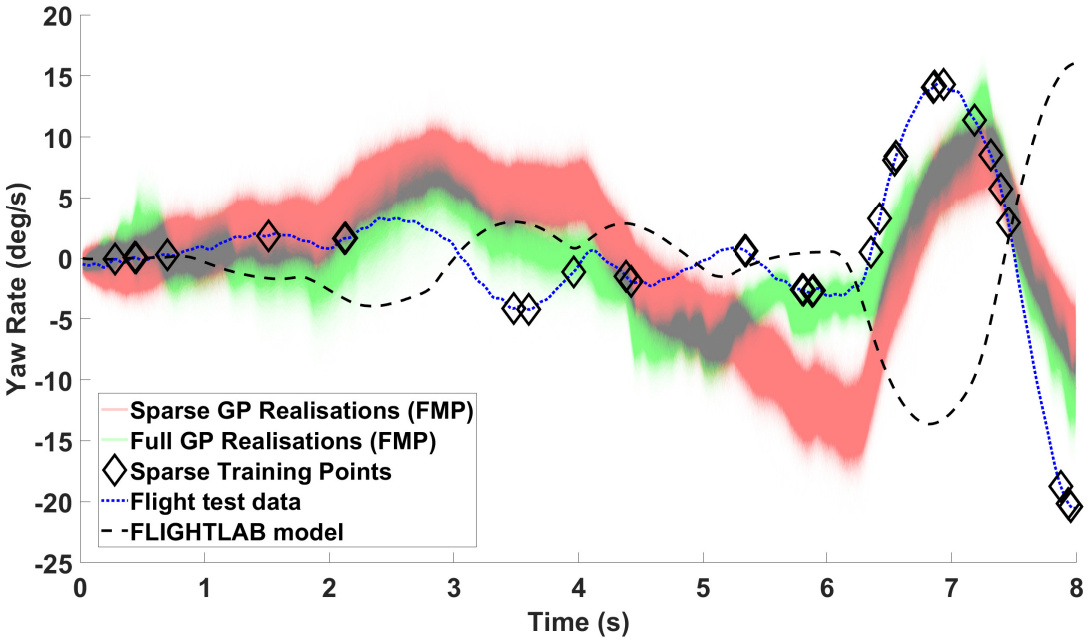


Figure 8.13: Realisations of sparse GP full model yaw rate predictions using 32 training points utilising one hyperparameter (α) (case 3, Table 8.1), with comparison to the corresponding full GP.

8.1.4 Heave

The convergence of the lower bound and hyperparameters for the sparse GP predicting the heave response, are shown in Figures 8.14 and 8.15, respectively. The α hyperparameter converges after 7 training points; however, the β hyperparameter does not converge until 12 training points. The lower bound is judged to have converged after 15 training points. In the current section, 32 training points are used to create the sparse GP model for the heave response (case 4, Table 8.1).

Figure 8.16 displays the rotorcraft inputs used for the input structure for the heave sparse GP. The black diamonds indicate the training points, which are automatically selected by the sparse GP approach. In Figure 8.16d the training points are concentrated around the highest and lowest points of the collective lever position. As previously discussed the values at the highest and lowest present the largest displacement in the input, therefore, should display the largest displacement in the output; this would provide more information about the dynamics of the system.

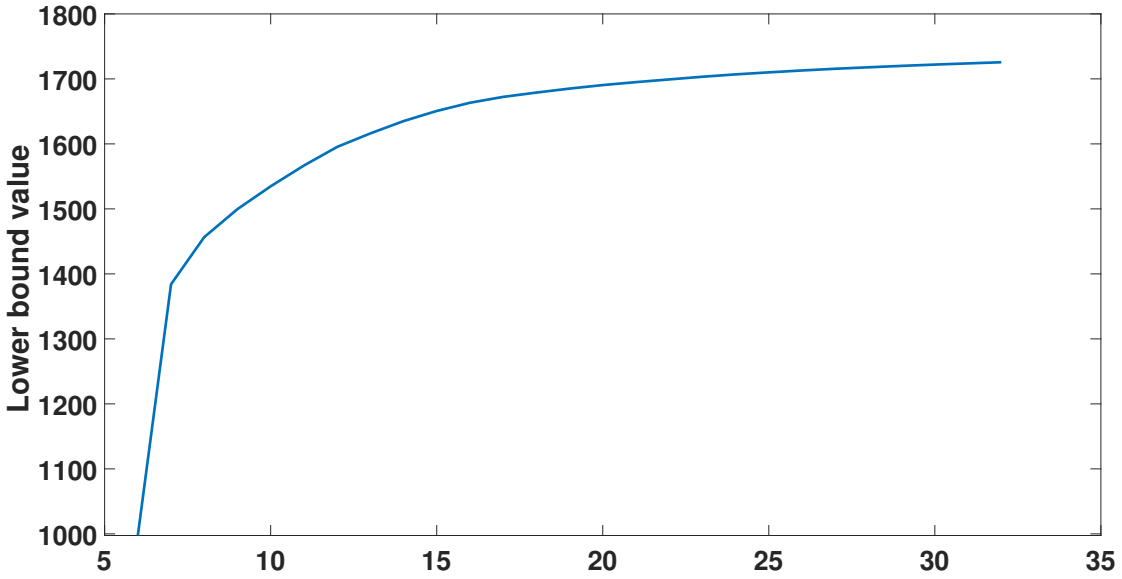


Figure 8.14: The convergence of the lower bound for the heave sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 4, Table 8.1).

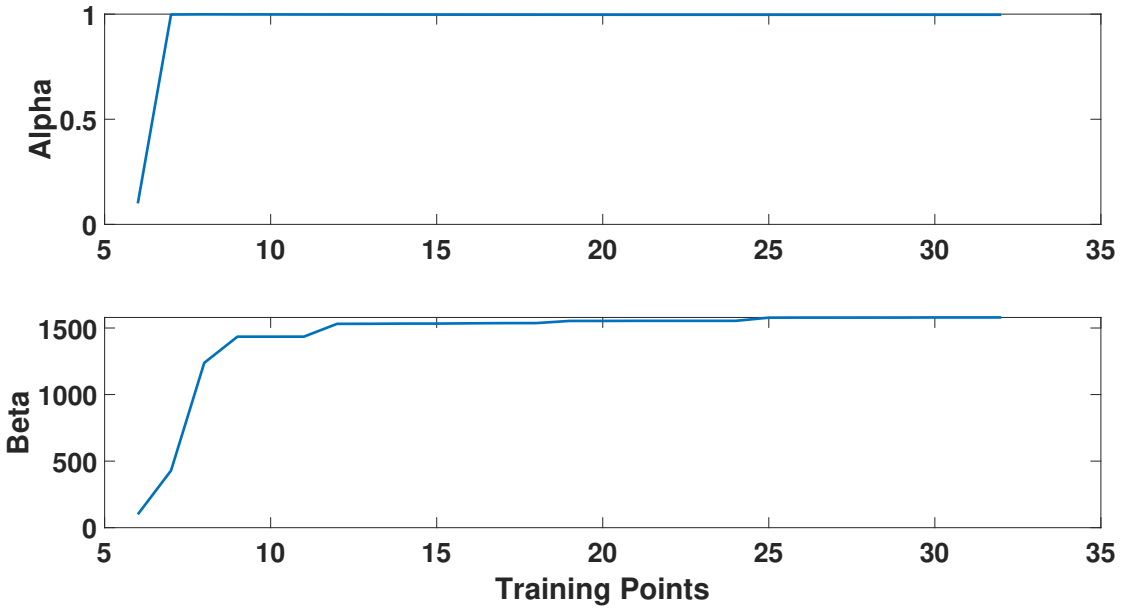
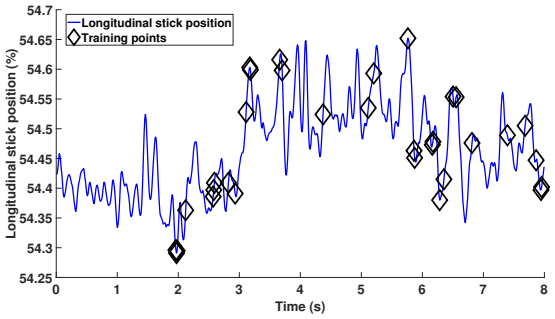
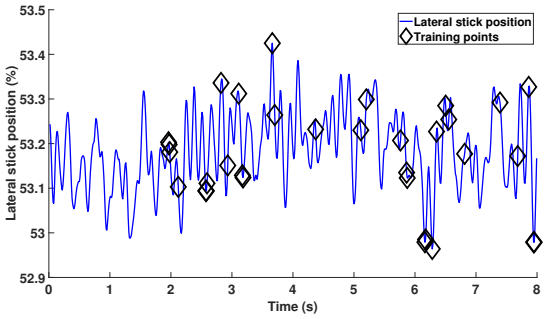


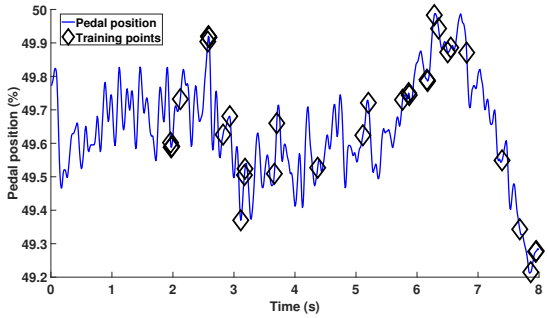
Figure 8.15: The convergence of the hyperparameters for the heave sparse GP prediction using 32 training points utilising one hyperparameter (α) (case 4, Table 8.1).



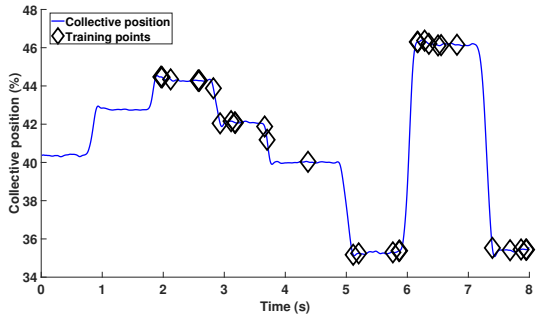
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 8.16: Bo105 rotorcraft inputs for the 3-2-1-1 collective input with 32 training points selected by the sparse GP approach.

Two sets of heave GP FMP realisations are given in Figure 8.17. The first set is the sparse GP FMP realisations using 32 training points and using the input/hyperparameter structure containing all of the current pilot inputs with one lagged heave term utilising one hyperparameter (case 4, Table 8.1) are shown by red realisations. The heave full GP is also shown in Figure 8.17 and displayed by green realisations. The sparse FMP realisations produce less accurate realisations compared to the full GP, however, both of the models outperform the FLIGHTLAB model. Note that, as previously discussed it is difficult to capture the dynamics of the heave response. Using the similarity measure for comparison between the GP predictions, the error between the sparse GP mean FMP realisations and the flight test data is 141.73, compared to the full GP, 34.45 and the corresponding FLIGHTLAB model 956.60. In this case, the sparse GP produces a significantly inferior model in terms of the similarity measure. The inferior sparse GP model could be due to the placement of the training points; in this case, the training points have been bunched in certain areas. In the pitch, roll and yaw rate sparse GP with 32 training points, the training points were spread across the time series.

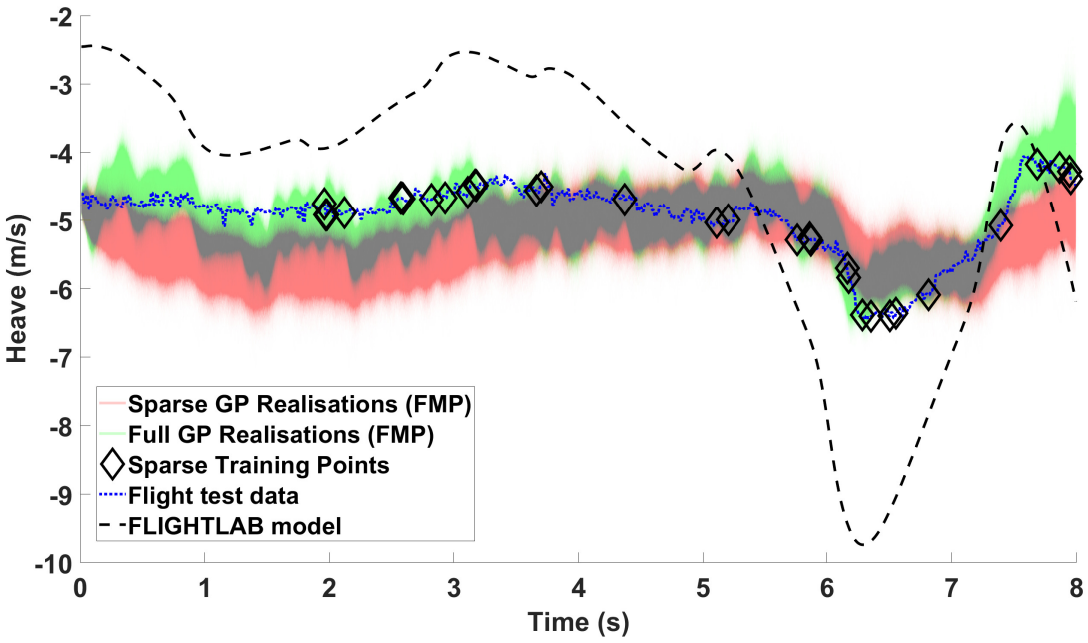
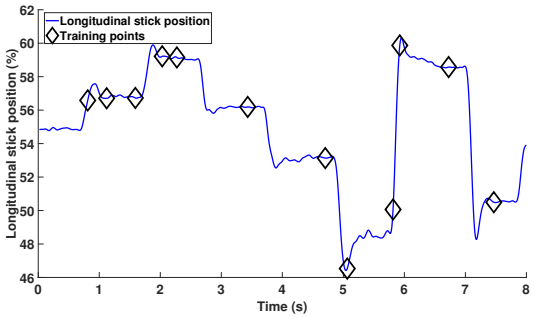


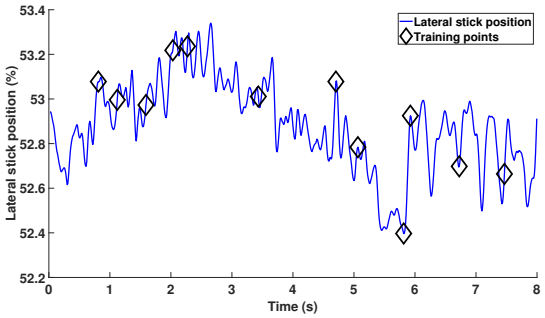
Figure 8.17: Realisations of sparse GP full model heave predictions using 32 training points utilising one hyperparameter (α) (case 4, Table 8.1), with comparison to the corresponding full GP.

8.2 Sparse Gaussian Process - Reduced Training Set

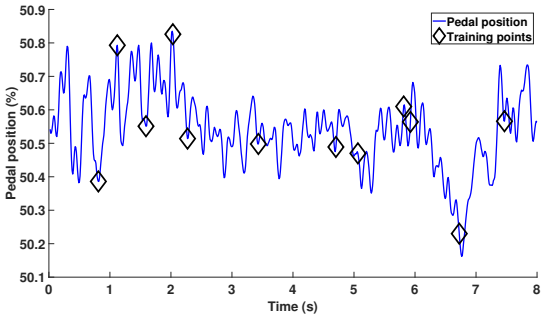
The results in the current section attempts to use a reduced training set to that of the full GP and to produce similar predictions to the full GP, based on the lower bound and hyperparameter convergence shown in Section 8.1. Using a reduced training set would have a positive impact on the time to train the GP model, primarily if a large dataset would be used, for example, a full flight test campaign. It would also be beneficial to speed up the predictions of the GP and potentially use the GP for real-time predictions. Each dataset, in which there are four, used in the current work have 800 data points; the sparse GP approach could potentially be used on a much larger dataset. The sparse GP approach would have the potential to significantly reduce the number of training points by attempting to select the optimal training points, which in turn would reduce the computational cost for training and prediction. Note that, if the GP model were going to be implemented into a flight simulator, it would have to be trained on a significantly larger dataset with more manoeuvres. It is, therefore, essential to show that the sparse approach can use a reduced number of training points and hence demonstrate the scalability to larger datasets.



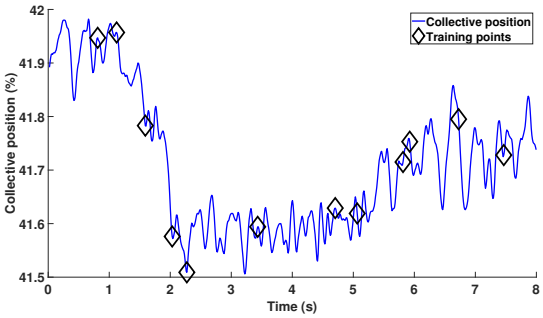
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 8.18: Bo105 rotorcraft inputs for the 3-2-1-1 longitudinal cyclic input with 12 training points selected by the sparse GP approach.

8.2.1 Pitch Rate

In the current section, an investigation is presented regarding the performance of the sparse GP when a reduced set of training points is utilised. As previously discussed in Section 8.1.1, 12 training points should provide enough information to produce accurate predictions. The OSAP figures are shown in Appendix H for the interested reader. The inputs used for the pitch rate GP are shown in Figure 8.18, where the black diamonds indicate the locations of the training points. In this case, 12 training points were automatically selected by the sparse GP approach. The 3-2-1-1 longitudinal excitation can be seen in Figure 8.18a, where it can be seen that the training points are spread across the time series. The location of the sparse points indicates that the information needed to make accurate predictions is not placed in the same locations in the manoeuvre, but spread across the manoeuvre.

Figure 8.19 displays the sparse GP FMP realisations for 12 training points, using the input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising one hyperparameter (case 1, Table 8.1). The use of a reducing training set, 12 in this case,

has had a detrimental impact on the FMP realisations, comparing the realisations to the full GP, there is a difference between the two models. The reason for the difference in the two models could be due to what has been previously discussed, the greedy selection algorithm and the challenges associated with predicting on FMP when the model is trained on OSAP. It can be seen in Figure 8.19 that from 6 seconds onwards the sparse GP struggles to capture the dynamics and does not encompass a number of training points. Not correctly predicting the training point is due to the FMP uncertainty propagation as the predictions are used in the input, this creates inputs which the model may not have been trained on; therefore, when the predictions reach 6 seconds onwards, the model has 'drifted' away from the training points. One thing to note is that even with a reduced training set and only using 12 training points, the sparse GP produces a more accurate model than the initial FLIGHTLAB model. The similarity measure for the sparse GP mean FMP realisations using 12 training points and the flight test data is 16.59, compared to the full GP, 5.62, and the corresponding FLIGHTLAB model 75.02.

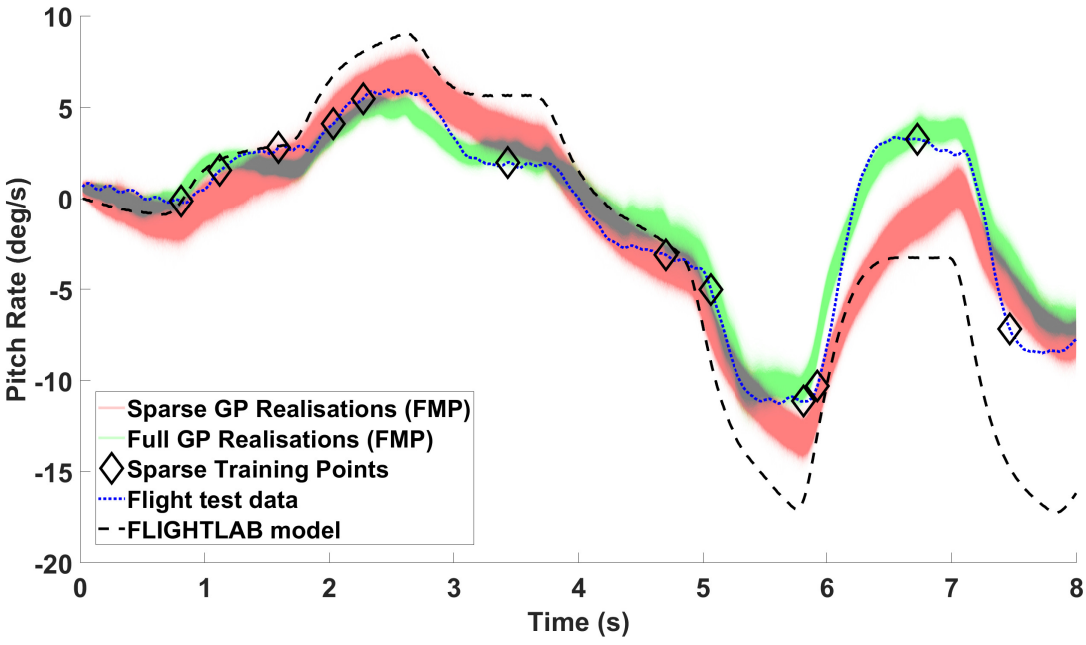
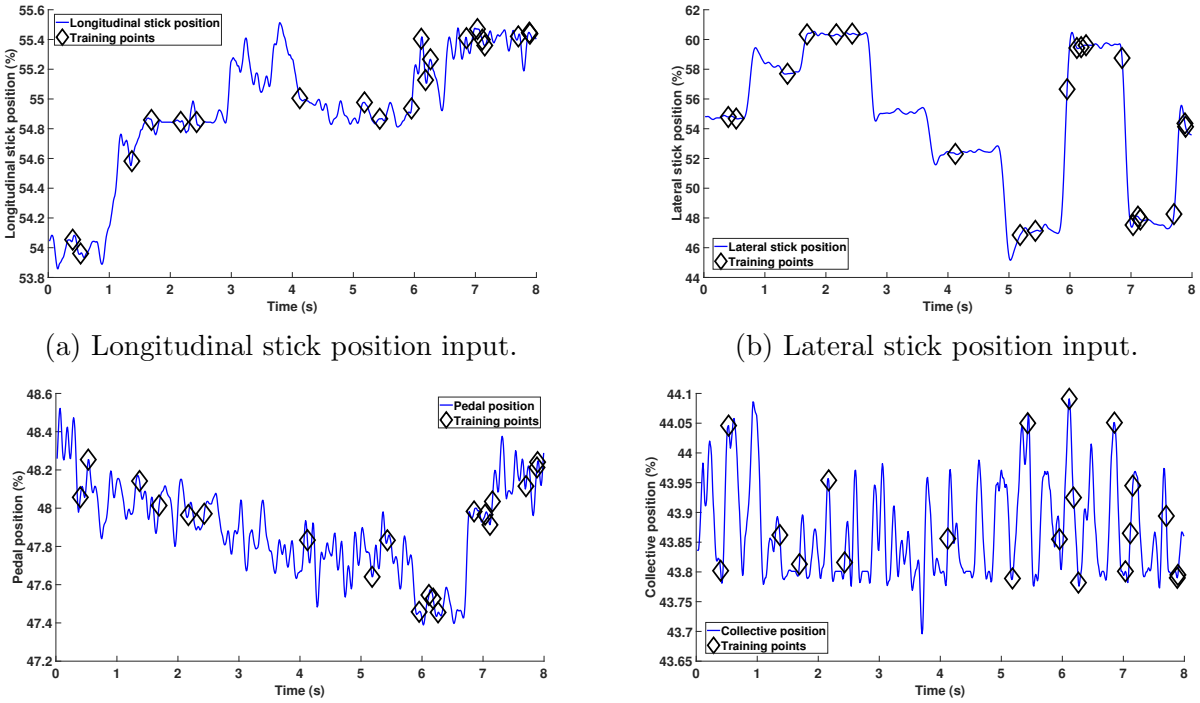


Figure 8.19: Realisations of sparse GP full model pitch rate predictions using 10 training points utilising one hyperparameter (α) (case 1, Table 8.1), with comparison to the corresponding full GP.



(a) Longitudinal stick position input.

(b) Lateral stick position input.

(c) Pedal position input.

(d) Collective position input.

Figure 8.20: Bo105 rotorcraft inputs for the 3-2-1-1 lateral cyclic input with 20 training points selected by the sparse GP approach.

8.2.2 Roll Rate

The roll rate sparse GP using a reduced training set is explored in the current section. In Section 8.1.2, the number of training points required to make accurate predictions is discussed, and it is stated that 20 training points should provide enough information to make accurate predictions for the roll rate response. The OSAP figures are shown in Appendix H. Figure 8.20 displays the inputs used for the roll rate sparse GP, where the black diamonds indicate the placement of the training points. The 3-2-1-1 lateral stick position excitation is shown in Figure 8.20b. As with the sparse GP pitch rate training points shown in the previous section, the training points are spread across the time series and not all located in one region.

The sparse GP FMP realisations for the roll rate response using 20 training points and the input/hyperparameter structure containing all of the current pilot inputs with two lagged roll rate terms utilising one hyperparameter (case 2, Table 8.1) is shown in Figure 8.21. The sparse GP FMP realisations are comparable to the mean FMP of the full GP, even with the use of a reduced training set. Visually it is not easy to distinguish which model produces

better predictions. It is, however, clear that both models outperform the FLIGHTLAB model. Quantifying the error of the sparse GP mean FMP realisations and the flight test data, the similarity measure is 9.62, compared to the full GP 10.30 and the corresponding FLIGHTLAB model 42.43. The results demonstrate that the sparse GP does create a marginally better model in terms of predictions.

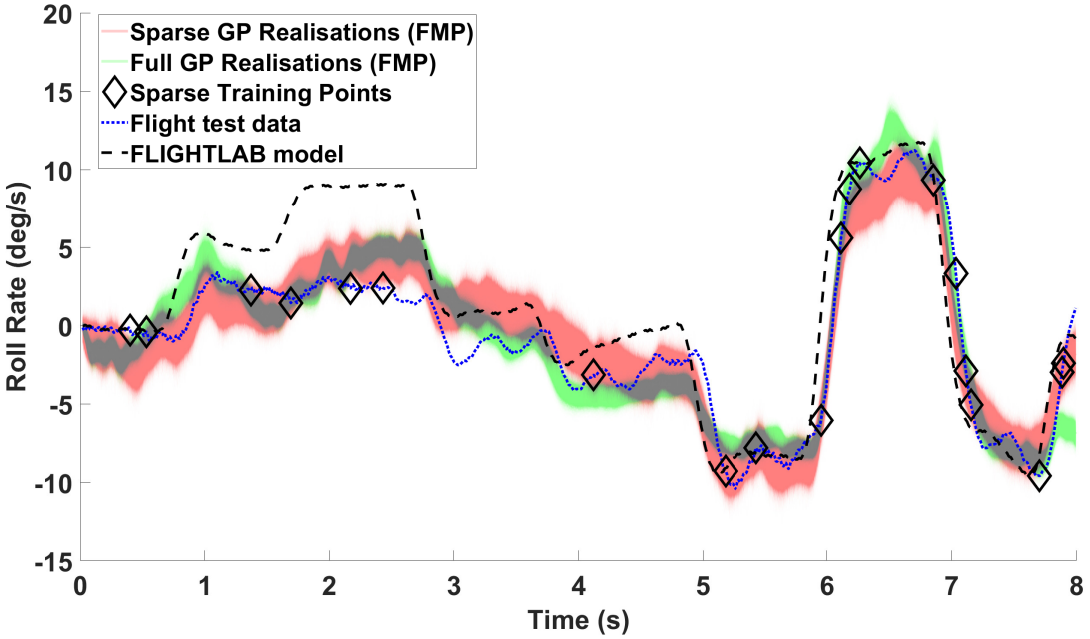
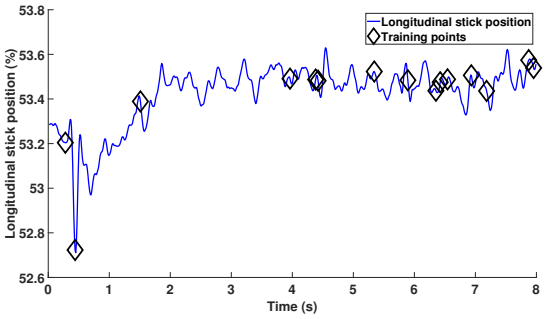


Figure 8.21: Realisations of sparse GP full model roll rate predictions using 10 training points utilising one hyperparameter (α) (case 2, Table 8.1), with comparison to the corresponding full GP.

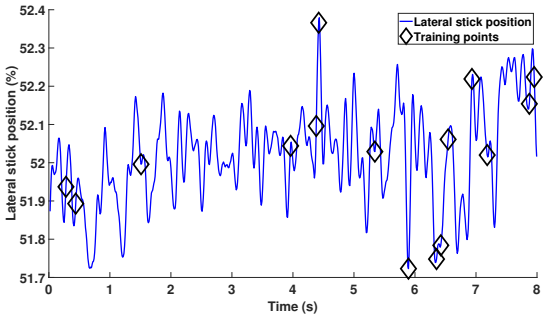
8.2.3 Yaw Rate

In the current section, the yaw rate response using less than 32 training points is investigated. In this case, as discussed previously in Section 8.1.3, 15 training points are used to create the yaw rate sparse GP. For the interested reader, the OSAP figures are shown in Appendix H. Figure 8.22 displays the input used for the yaw rate sparse GP. The 3-2-1-1 excitation of the pedal position is shown in Figure 8.22c; the training points are selected across the time series.

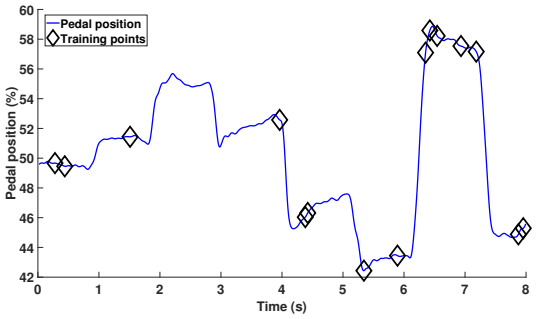
Figure 8.23 shows the sparse GP FMP realisations of the yaw rate response using 15 training points and the input/hyperparameter structure containing all of the current pilot inputs with two lagged yaw rate terms utilising one hyperparameter (case 3, Table 8.1). The sparse GP FMP realisations in Figure 8.23, produce less accurate realisations than the FMP realisations



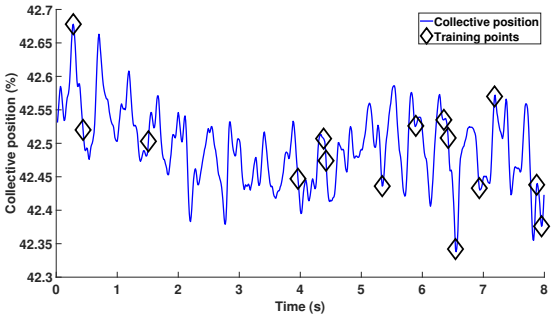
(a) Longitudinal stick position input.



(b) Lateral stick position input.



(c) Pedal position input.



(d) Collective position input.

Figure 8.22: Bo105 rotorcraft inputs for the 3-2-1-1 pedal input with 15 training points selected by the sparse GP approach.

of the full GP. Quantifying the error of the sparse GP mean FMP realisations and the flight test data, the similarity measure is 171.06, compared to the full GP 53.26 and the corresponding FLIGHTLAB model 370.08. In this case, both GP models outperform the FLIGHTLAB model, with the full GP producing a more accurate model in terms of the response dynamics. The reason around this could be due to what has previously discussed; a random selection of the points during the greedy optimisation.

8.2.4 Heave

In the current section, the use of sparse GP for the heave response using a reduced training set is investigated. As discussed in Section 8.1.4, the use of 15 training points will be used in the current analysis. Figure 8.24 displays the rotorcraft inputs used for the heave response sparse GP. The excitation of the collective lever with the 3-2-1-1 manoeuvre is displayed in Figure 8.24d; it can be seen that there are no training points selected in the first 2 seconds of the input. The selection of the training points would suggest that the first 2 seconds of the input does not contain highly informative information. However, there are training points at

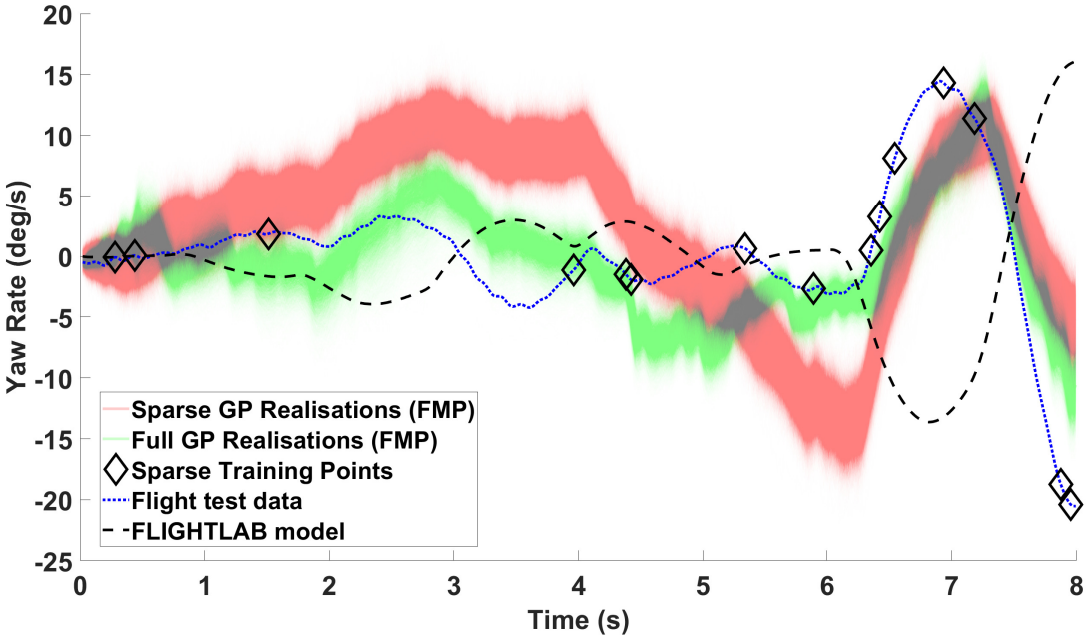
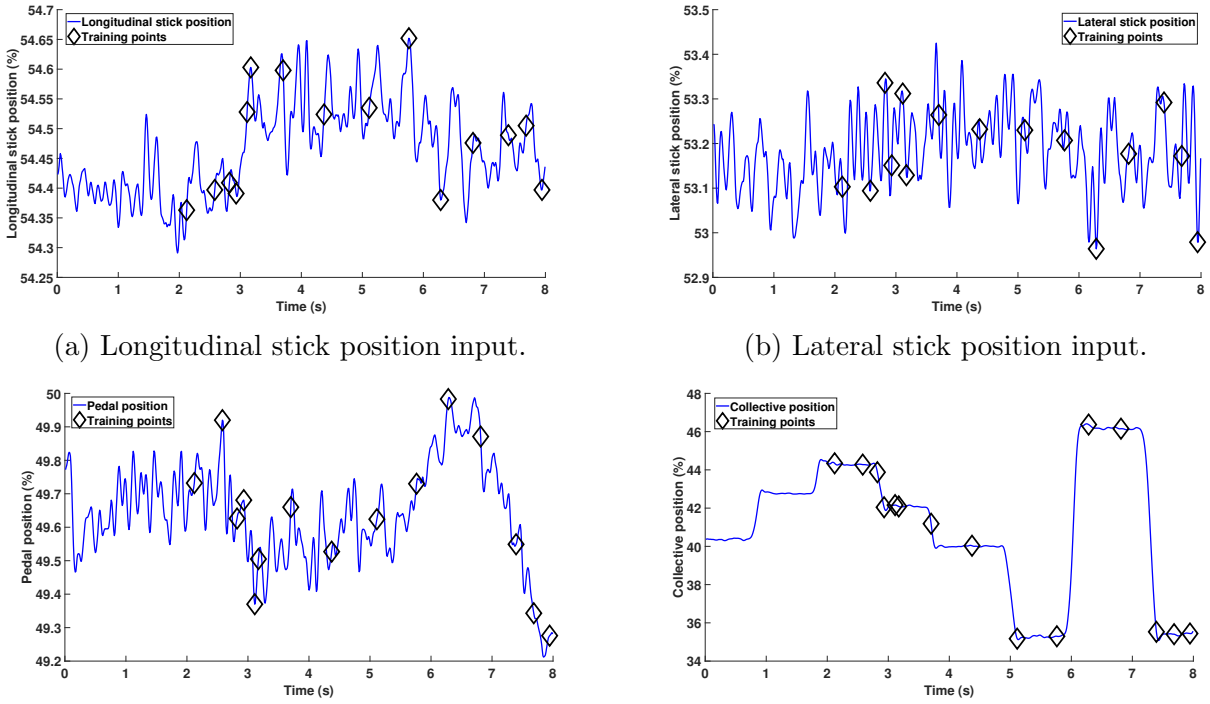


Figure 8.23: Realisations of sparse GP full model yaw rate predictions using 10 training points utilising one hyperparameter (α) (case 3, Table 8.1), with comparison to the corresponding full GP.



(a) Longitudinal stick position input.

(b) Lateral stick position input.

(c) Pedal position input.

(d) Collective position input.

Figure 8.24: Bo105 rotorcraft inputs for the 3-2-1-1 collective input with 15 training points selected by the sparse GP approach.

the highest and lowest input values; the reason for this has already been discussed.

The sparse GP FMP realisations for the heave response using 15 training points and the input/hyperparameter structure of all of the current pilot inputs with one lagged heave term utilising one hyperparameter (case 4, Table 8.1) is shown in Figure 8.25. The FMP realisations are very similar to the full GP, this is more evidence that the sparse GP, even with a smaller number of training points can create a comparable model to that of the full GP. Confirming this with similarity measure, the value between the sparse GP mean FMP realisations and the flight test data is 31.01, compared to the full GP 34.45 and the FLIGHTLAB model 956.60.

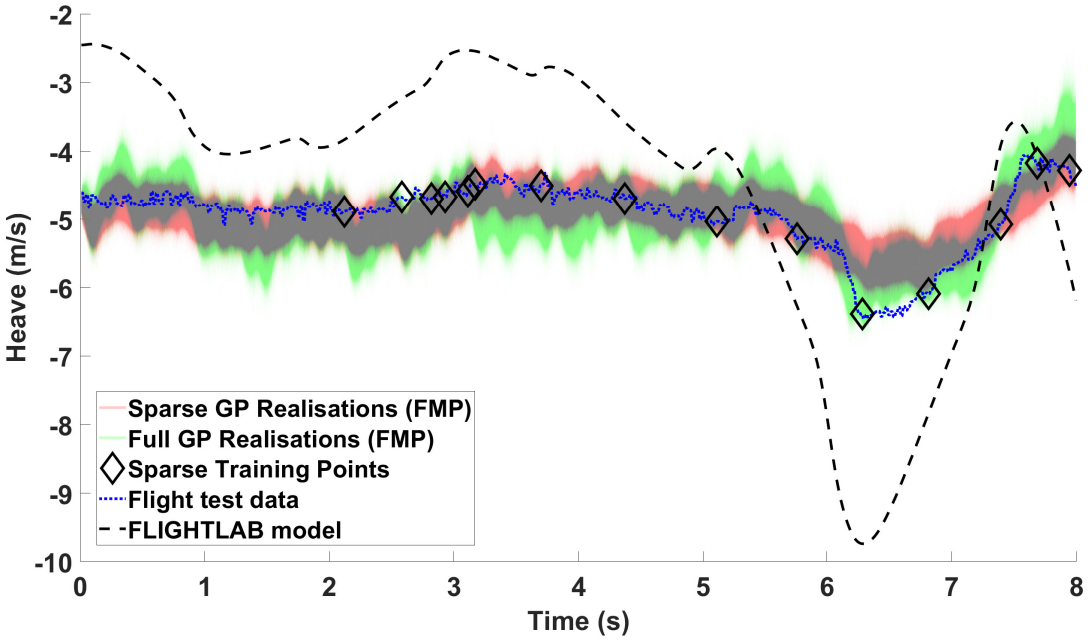


Figure 8.25: Realisations of sparse GP full model heave predictions using 15 training points utilising one hyperparameter (α) (case 4, Table 8.1), with comparison to the corresponding full GP.

8.3 Summary

As discussed in the chapter, the reason the sparse GP may not be as accurate in terms of predicting the model response is based on the hypothesis that restricting the sparse points to be part of the dataset has had a negative effect on the performance due to the greedy selection algorithm. For future work, one could investigate points that are not part of the dataset to see if there is improved performance benefits, as described in [65].

The similarity measure for all of the GP models using 32 training points is shown in Table 8.2. Note that all of the GP models utilising 32 training points outperform the corresponding

FLIGHTLAB model. The test of the sparse GP was to create a comparable model to that of the full GP, in the case of pitch and roll rate. However, for the heave and yaw rate response, the sparse GP using 32 training points is significantly worse than the full GP. One hypothesis is that the OSAP sparse GP prediction shown in Appendix H, specifically Figure H.4, may be overfitted.

The similarity measure for the sparse GP models that used a reduced training set is shown in Table 8.2, where the pitch rate used 12 training points, roll rate used 20 training points, and the yaw rate and the heave response used 15 training points. As with the results shown in Section 8.1, the sparse GP models outperform the corresponding FLIGHTLAB models. In this case, the sparse GP created predictions that are comparable to the full GP using a reduced training set and even outperforms the full GP for the roll rate and heave response. For the pitch rate response, the sparse GP model using 12 training points is worse than the full GP; however, given the reduced time complexity and the fact that it still outperforms the initial FLIGHTLAB model, it would still be beneficial to use the reduced training set. The results shown in the current chapter have provided evidence that the sparse GP can create a model with predictions that are similar or better than the full GP using a reduced training set for the pitch rate, roll rate and heave responses. The yaw rate response does not create a sparse GP model that is comparable to the full GP. It could be due to the reasons already discussed, which include the model being training on OSAP to predict FMP and the greedy selection method.

The reduction in training points would significantly reduce the cost of applying the proposed method to much larger datasets in terms of the time to train the model. In terms of scalability, the variational sparse GP approach would, therefore, be more practical for future applications with larger data sets, as the results presented have provided evidence that the variational sparse GP approach can reduce the number of training points, thus reducing the time required for training.

Using the reduced training set of the sparse GP models increased the potential for the GP to be able to run in real-time. For the GP using 10 training points, one full set of realisations which was 8 seconds in time (800 predictions) takes between 0.03-0.045 seconds and the for GP using 15 training points between 0.035 - 0.05 seconds. The time to make predictions displays

evidence that the GP can run in real-time for flight simulator predictions.

Similarity Measure for on-axis response Sparse GP				
Dataset	**Prediction**	**Case**	**Model**	**Similarity Measure**
3-2-1-1 Longitudinal Input	Pitch Rate	1	FMP full GP	**5.62**
			FMP Sparse GP 32 Points	**8.54**
			FMP Sparse GP 12 Points	**16.59**
			FLIGHTLAB	**75.02**
3-2-1-1 Lateral Input	Roll Rate	2	FMP full GP	**10.30**
			FMP Sparse GP 32 Points	**15.29**
			FMP Sparse GP 20 Points	**9.62**
			FLIGHTLAB	**42.43**
3-2-1-1 Pedal Input	Yaw Rate	3	FMP full GP	**53.26**
			FMP Sparse GP 32 Points	**129.45**
			FMP Sparse GP 15 Points	**171.06**
			FLIGHTLAB	**370.08**
3-2-1-1 Collective Input	Heave	4	FMP full GP	**34.45**
			FMP Sparse GP 32 Points	**141.73**
			FMP Sparse GP 15 Points	**31.01**
			FLIGHTLAB	**956.60**

Table 8.2: Similarity measure for on-axis response Sparse GP results

Chapter 9

Investigation of Model Error using Machine Learning

Model error captures the inaccuracy of the model compared to the real system (see studies by Kennedy and O'Hagan [72], Goldstein and Rougier [73] and Brynjarsdóttir and O'Hagan [74]). The studies by O'Hagan attempt to 'tune' the model parameters alongside creating a model to predict model error. In the current chapter, the author investigates creating a model that predicts the model error of an already 'tuned' FLIGHTLAB model. As previously discussed in Chapter 3, the model error approach is one method of implementing the GP model into a flight simulator, where in the GP model attempts to correct the physical law based FLIGHTLAB. The model error of the pitch rate response from the 3-2-1-1 manoeuvre exciting the longitudinal stick position is shown in Figure 9.1, where the red line displays the discrepancy between the model and the real system. A black dashed line shows the prediction from the FLIGHTLAB model, which has been tuned by an 'expert' simulation engineer from the University of Liverpool flight science and technology research group. The flight test data is displayed by the blue dashed line. The model error is defined as:

$$\boldsymbol{\rho} = \boldsymbol{y}^{\boldsymbol{fl}} - \boldsymbol{y} \tag{9.1}$$

where $\boldsymbol{\rho}$ is the model error, $\boldsymbol{y}^{\boldsymbol{fl}}$ is the physical law based FLIGHTLAB model, and $\boldsymbol{y}$ is the

Bo105 flight test data. Both the datasets are described in Chapter 3.

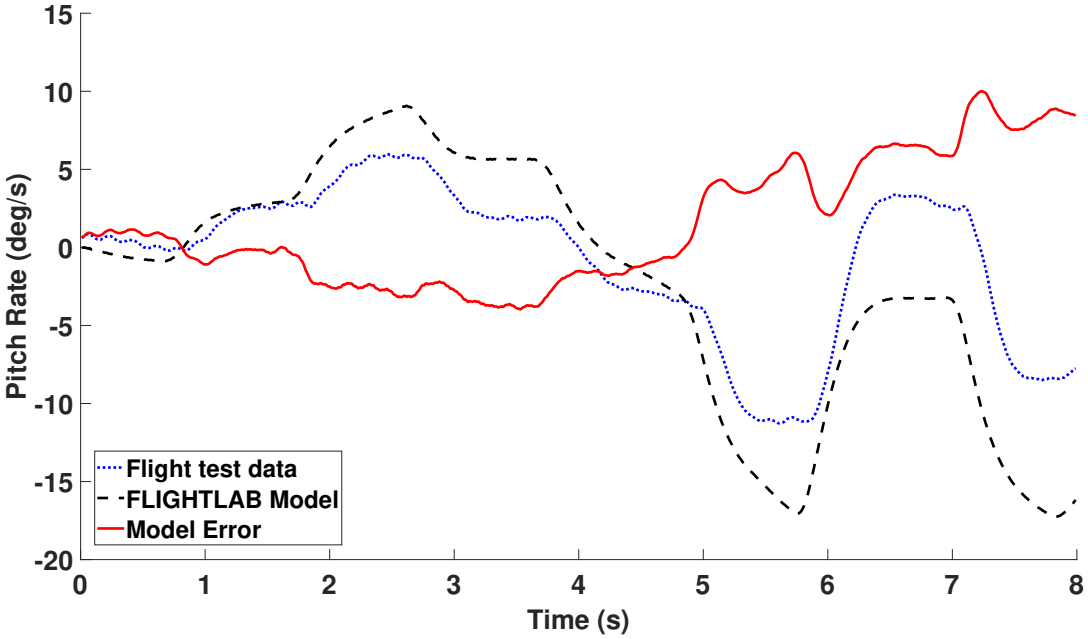


Figure 9.1: An example of the pitch rate model error.

The concept of model error is that the machine learning regression would be able to predict the discrepancy ($\boldsymbol{\rho}$), and creates what is known as a 'grey-box' model (FLIGHTLAB + regression model for predicting model error). This idea may be well suited to a system such as a rotorcraft flight simulator, where a model already exists. Equation (9.2) shows the resulting grey-box model

$$y^*(x_i) = f(x_i) + R(x_i) + \epsilon, \quad i = 1, \ldots, N \tag{9.2}$$

where N is the number of predictions, x are the pilot input, R is a machine learning regression model predicting the model error($\rho(x_i)$) and y^* is the predictions made by the grey-box model.

In the current chapter, the use of machine learning to predict model error is investigated to see if it can improve the accuracy of the flight simulators (which are called FLIGHTLAB models in the current thesis). The model error method has the ability to be able to correct the FLIGHTLAB model, combining a data-based machine learning model with the physical law based FLIGHTLAB model, creating a grey-box model. It is hypothesised that the grey-box

approach has the potential to improve the predictions for the flight simulator and that it may be more robust in regions where there is little or no data. In the current chapter, it is investigated if the grey-box model can improve on the FLIGHTLAB predictions for the pitch rate, roll rate, yaw rate, and heave.

9.1 Model Error Input Structures

The input structures used are outlined in Chapter 4; however, in the case of the current chapter, the machine learning regression model is predicting the model error ($\boldsymbol{\rho}$). For this reason, it is beneficial to outline the input structures again. In the current section, the input structure for the pitch rate response of the Bo105 rotorcraft from 3-2-1-1 longitudinal manoeuvre are shown. Firstly, the moving average (MA) input structure is a linear combination of past inputs. In the case of the pitch rate response of the Bo105 rotorcraft from the 3-2-1-1 longitudinal manoeuvre, the input is the longitudinal stick position (δ^x):

$$\boldsymbol{x}_n = \begin{pmatrix} \delta^x_{n-1} \end{pmatrix} \tag{9.3}$$

where n represents the point in the time history of the Bo105 flight test data. The auto-regressive input structure contains lagged previous outputs, in this case, the output is the model error pitch rate. The input structure is therefore of the form

$$\boldsymbol{x}_n = \begin{pmatrix} \rho_{n-1} \end{pmatrix}. \tag{9.4}$$

The auto-regressive with exogenous inputs (ARX) input structure builds on the AR input structure by including the current input. In the case of the pitch rate response from Bo105 rotorcraft (for the 3-2-1-1 longitudinal manoeuvre), the current input is the longitudinal stick position. The ARX enables the model to capture the input and output of the Bo105 rotorcraft. The ARX input structure is

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \\ \rho_{n-1} \end{pmatrix}. \tag{9.5}$$

Combining two of the previously discussed input structures, AR and MA, produces an auto-regressive moving average (ARMA) model. Application to the Bo105 rotorcraft uses a lagged model error pitch rate response and a lagged longitudinal stick position, such that

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_{n-1}^x \\ \\ \rho_{n-1}^x \end{pmatrix}. \tag{9.6}$$

Finally, combining the ARX and MA input structures creates the auto-regressive moving average with exogenous inputs (ARMAX) input structure. In the context of the Bo105 rotorcraft the ARMAX input structure contains the current and one lagged longitudinal stick position with one lagged model error pitch rate response:

$$\boldsymbol{x}_n = \begin{pmatrix} \delta_n^x \\ \\ \delta_{n-1}^x \\ \\ \rho_{n-1} \end{pmatrix}. \tag{9.7}$$

9.2 Linear Regression Similarity Measure

The first test is to investigate if a linear regression model can capture the dynamics of the model error pitch rate. Table 9.1 displays the similarity measures for the OSAP from the linear regression models that utilised five different input structures; the input structures are shown in Section 9.1. The OSAP of the linear regressions models are not accurate; therefore, it was decided not to display the figures in the main text; however, these can be seen in Appendix I.1. Due to the poor performance of the OSAP, it was also decided not to undertake the FMP as this is a more stringent test of the model performance. Using the results in Table 9.1, the ARMAX input structure produces the more accurate model in terms of OSAP; however, this is not enough evidence to state that the ARMAX input structure would also produce the most

accurate results for the GP models. In the next section, the aforementioned input structures are utilised in a GP model to display which input structure produces the most accurate predictions of model error for the pitch rate response.

Prediction	**Input Structure**	**Similarity measure**
Pitch rate	Moving Average (MA)	65.14
	Auto-regressive (AR)	43.03
	Auto-regressive with exogenous inputs (ARX)	9.26
	Auto-regressive moving average (ARMA)	9.25
	Auto-regressive moving average with exogenous inputs (ARMAX)	5.55

Table 9.1: The similarity measures for the pitch rate OSAP from linear regression models using different input structures.

9.3 Investigation of the Gaussian Process Input Structure

In the current section, the input structures that are outlined in Section 9.1 are used to create GP models to predict the pitch rate response model error to find which input structure creates the most accurate predictions. Using the results outlined in the previous section, it was shown that the input structures moving average and auto-regressive created models that were not as accurate; therefore, they will not be investigated in the current section. In this section, the simulated annealing results associated with the GP models are not displayed, but for the interested readers are shown in Appendix I.2. For the sake of brevity, the results for OSAP are also not shown in the current section but are shown in Appendix I.2.

Figure 9.2 displays three sets of realisations for GP models, using different input structures, predicting the model error pitch rate. The red realisations display the FMP for the GP model using an auto-regressive with exogenous input structure (equation (9.5)), the green realisations

show the FMP for the GP model using an auto-regressive moving average input structure (equation (9.6)) and the light blue realisations display the FMP for the GP model using an auto-regressive moving average with exogenous input structure (equation (9.7)). The first thing to note is that the input structures of ARX and ARMA produced very similar models, as the red and green realisations create the grey area in Figure 9.2. The ARMAX input produces realisations that look more accurate than the ARX and ARMA GP models between 6 to 7 seconds, however, in the middle part of the time series the ARMAX GP model creates realisations that are far from the true model error. Note that, depending on 'trajectory' taken, the realisations seem to be able to deviate onto a second set of grouped solutions that is far from the training data, this error can not be highlighted when only using the OSAP.

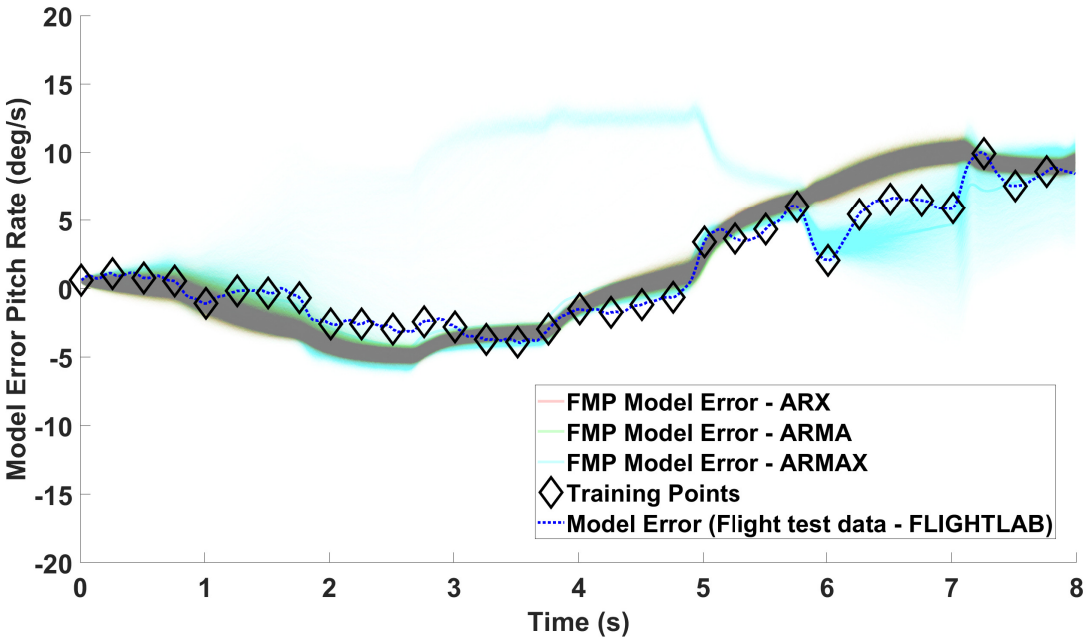


Figure 9.2: Model error pitch rate FMP realisations from three GP models utilising an auto-regressive with exogenous input structure, an auto-regressive moving average input structure and an auto-regressive moving average with exogenous input structure.

The grey-box model (GP predicting model error + FLIGHTLAB model) for the three GP models (ARX, ARMA and ARMAX) is displayed in Figure 9.3. The colours of the realisations are the same as Figure 9.2, and also displays similar results. That the ARX and ARMA GP model produces very similar realisations and that the ARMAX model is the most accurate between 6 to 7 seconds, however, produces realisations that are far from the flight test data in the middle part of the time-series.

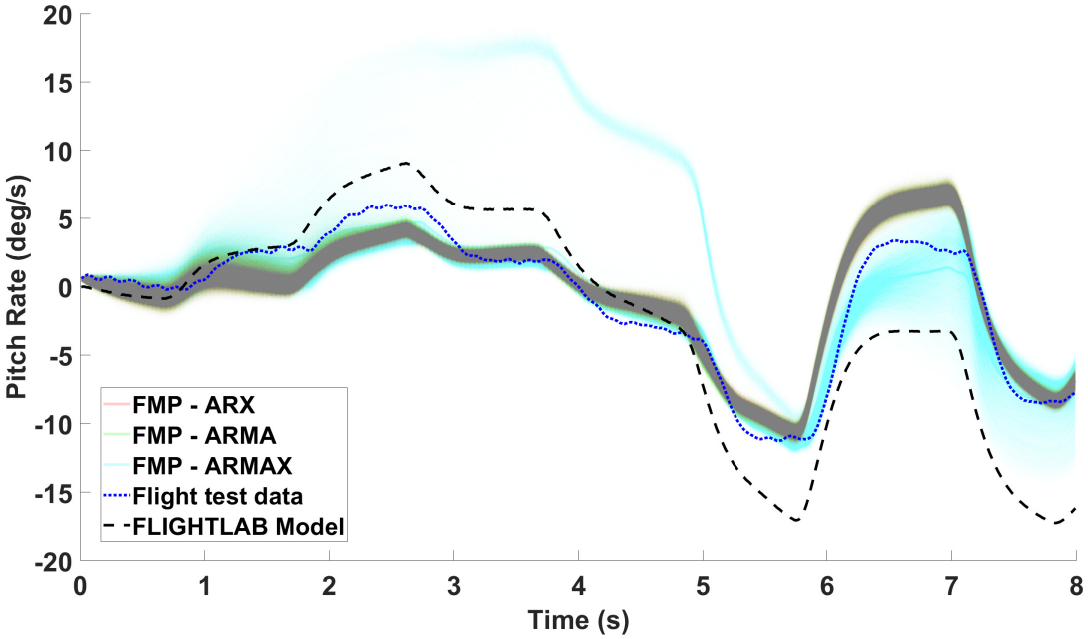


Figure 9.3: Pitch rate grey-box FMP realisations from three GP models utilising an auto-regressive with exogenous input structure, an auto-regressive moving average input structure and an auto-regressive moving average with exogenous input structure.

Quantifying the accuracy of the model in terms of similarity measure between the mean of the GP FMP realisations and the flight test data, the ARX GP model has a similarity measure of 14.00, the ARMA GP model has a similarity measure of 13.65, and the ARMAX GP model has a similarity measure of 6.32. The results are displayed in Table 9.2, alongside the OSAP similarity measure. Using the similarity measure the GP model that utilises the ARMAX input structure produces the most accurate FMP. However, it can be seen in Figures 9.2 and 9.3 that the ARMAX GP model produces FMP realisations that are far from the truth. The reason the similarity measure does not reflect the divergent behaviour of the FMP is due to calculation of the similarity measure which utilises the mean of the FMP realisations.

Prediction	**Input Structure**	**GP Prediction**	**Similarity measure**
Model error Pitch rate	Auto-regressive with exogenous inputs (ARX)	OSA	0.03
		FMP	14.00
	Auto-regressive moving average (ARMA)	OSA	0.03
		FMP	13.65
	Auto-regressive moving average with exogenous inputs (ARMAX)	OSA	0.02
		FMP	6.32

Table 9.2: The similarity measures for the pitch rate predictions from linear regression models using different input structures.

It is also beneficial to investigate the accelerations from the three grey-box models. Figure 9.4 shows the predicted accelerations of the grey-box models. It can be seen that the ARX and ARMA models capture the acceleration better than ARMAX, and the ARMAX model has larger extreme values than the GP models utilising the ARX and ARMA input structures.

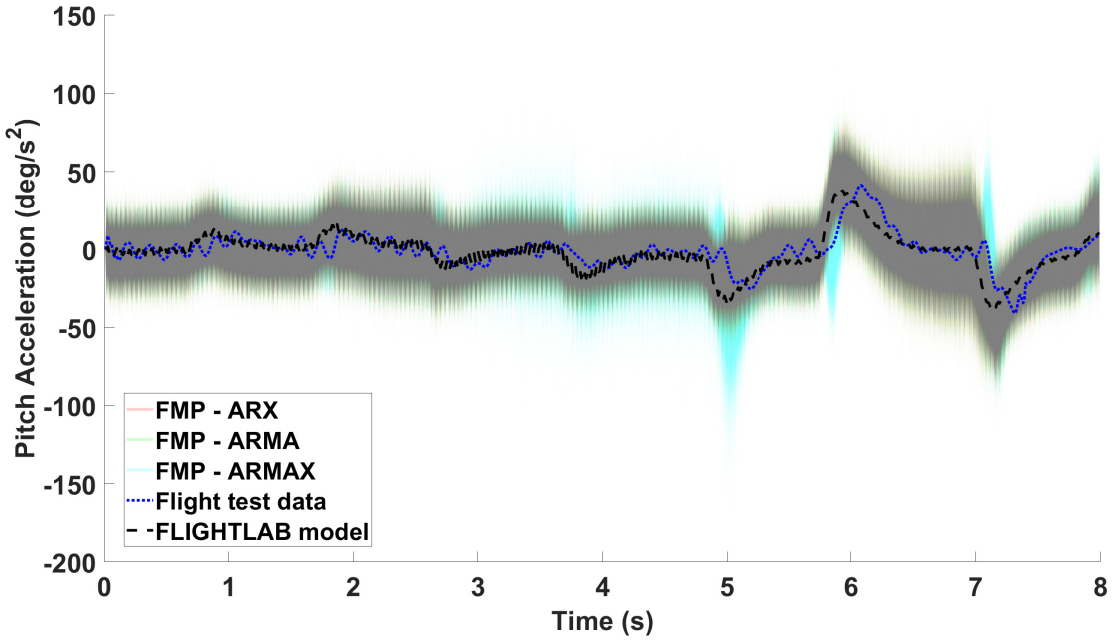


Figure 9.4: Acceleration of the pitch rate grey-box full model realisations from three GP models utilising an auto-regressive with exogenous input structure, an auto-regressive moving average input structure and an auto-regressive moving average with exogenous input structure.

In the current section, it was shown that using the ARMAX input structure produced the most accurate results for predicting pitch rate in terms of the similarity measure. However, it was shown in terms of accelerations, the ARX and ARMA models produced more accurate results than the GP utilising the ARMAX input structure. It is important for the acceleration of the prediction to closely match that of the true rotorcraft acceleration, as the more accurate the acceleration of the prediction is, the closer the flight simulator will 'feel' like the true rotorcraft. Due to the results displayed in the current section, it is not beneficial to use ARMAX. The ARX/ARMA GP models produced very similar results. It was decided that an ARX input structure was to be used to predict model error in the remainder of the chapter as this was used for the GP results previously displayed in the thesis. Taking knowledge from previous chapters, it is known that utilising a hyperparameter for each input in the GP input structure does not always improve the predictions. Using more than one hyperparameter increases the

dimensionality of the optimisation problem, and the optimal solutions may not be reached, as discussed in Chapter 6. For this reason, in the case of model error, the current chapter only utilises one hyperparameter (α), including the noise precision (β).

9.4 Gaussian Process Models Utilising an ARX Input Structure

In the current section, the ARX models are investigated for the pitch rate, roll rate, yaw rate, and heave GP models. For each response, four different ARX input structures are investigated and are displayed in Table 9.3 as cases. The input structure alters the pilot current inputs and the number of lagged model error terms. For consistency, the GPs shown in Chapter 5 are referred to as full GPs. The pitch rate, roll rate and yaw rate GPs were created with an input structure containing all of the current pilot inputs and two lagged response terms. The heave GP was created using an input structure containing all of the current pilot inputs and one lagged heave response term.

Manoeuvre	Prediction	Pilot input	Lagged Output	Case
3-2-1-1 Longitudinal input	Pitch rate	Current longitudinal pilot input	One model error pitch rate term	1
			Two model error pitch rate terms	2
		All of the current pilot inputs	One model error pitch rate term	3
			Two model error pitch rate term	4
3-2-1-1 Lateral input	Roll rate	Current lateral pilot input	One model error roll rate term	5
			Two model error roll rate terms	6
		All of the current pilot inputs	One model error roll rate term	7
			Two model error roll rate terms	8
3-2-1-1 Pedal input	Yaw rate	Current pedal pilot input	One model error yaw rate term	9
			Two model error yaw rate terms	10
		All of the current pilot inputs	One model error yaw rate term	11
			Two model error yaw rate terms	12
3-2-1-1 Collective input	Heave	Current collective pilot input	One model error heave term	13
			Two model error heave terms	14
		All of the current pilot inputs	One model error heave term	15
			Two model error heave terms	16

Table 9.3: GP configurations for the investigation of the GP model error

9.4.1 Pitch Rate

The pitch rate response is investigated first. Figure 9.5 displays the pitch rate response of the flight test data from the Bo105 rotorcraft along with the corresponding FLIGHTLAB model attempting the same manoeuvrer, which in the case of the pitch rate is a 3-2-1-1 longitudinal stick position (cases 1, 2, 3 and 4 Table 9.3). The model error is displayed by the red line, to show the error between the flight test data and the FLIGHTLAB model. The model error shown is the response that the GP attempts to predict (such that the GP will attempt to account for discrepancies in the FLIGHTLAB predictions).

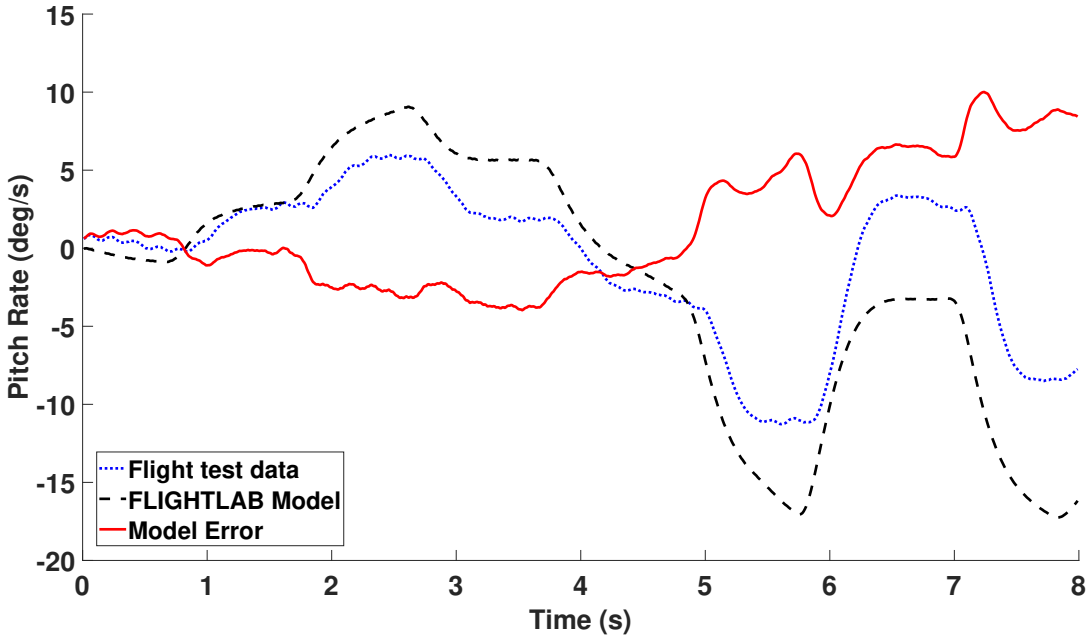


Figure 9.5: Model Error for the pitch rate response (Flight test data - FLIGHTLAB model).

Firstly, different GP models using ARX input structures are investigated to identify which creates the most accurate predictions. The four input structures used were longitudinal stick position with one/two lagged model error pitch rate terms (case 1 and 2, Table 9.3) and all of the current pilot inputs with one/two lagged model error pitch rate terms (case 3 and 4, Table 9.3). The two sets of GP FMP realisations for each input structure, using one lag term, attempting to predict the model error pitch rate are displayed in Figure 9.6 and are both trained on 32 points (selected to be every 25th point in the dataset). The red realisations show the input structure containing the longitudinal stick position with one lagged model error pitch

rate term, and the green realisations display the input structure containing all of the current pilot inputs with one lagged model error pitch rate term.

For the sake of brevity, the simulated annealing and OSAP figures are displayed in Appendix I.3.

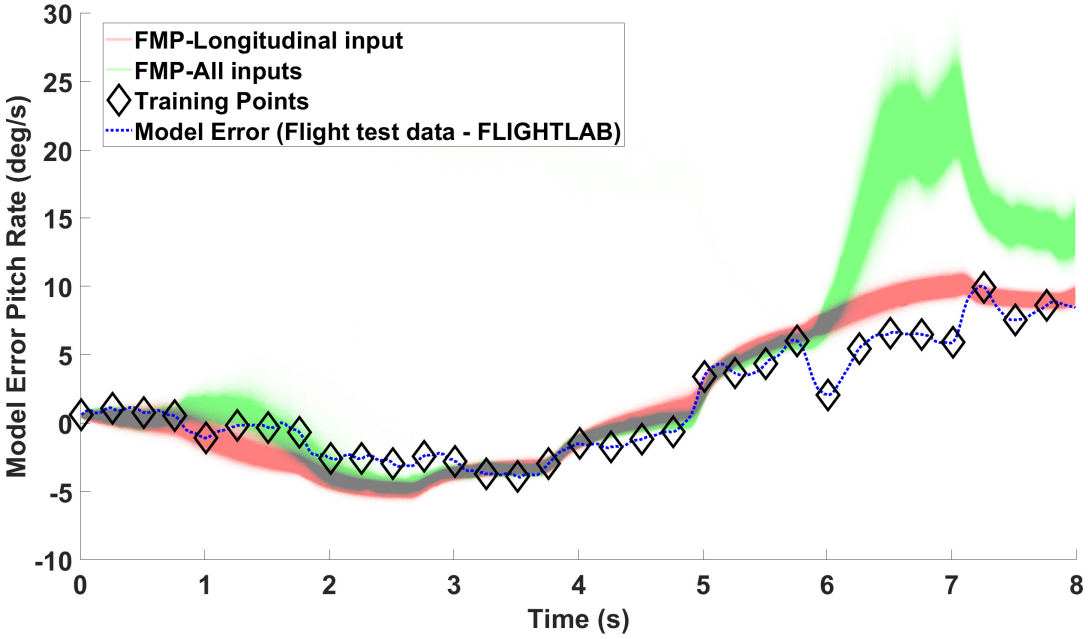


Figure 9.6: Comparison of the GP FMP realisations using different input structures (cases 1 and 3, Table 9.3) to predict the model error pitch rate.

Figure 9.7 displays two sets of FMP realisations, both trained on 32 training points. The red realisations show the results obtained when using the input structure containing the longitudinal stick position with two lagged model error pitch rate terms, while the green realisations display the results that are obtained when using input structure containing all of the current pilot inputs with two lagged model error pitch rate terms.

Visually, using Figures 9.6 and 9.7, it is clear that the FMP realisations for case 1 and 2 (red realisations) outperform the FMP realisations for case 3 and 4 (green realisations). The result is not what is expected, as it would be more feasible to assume that the input structure containing all of the current pilot inputs would produce better or similar results to the input structure containing the current longitudinal pilot input. The use of all the pilot current inputs in the GP input structure provides evidence that the other pilot inputs (lateral stick position, pedal position and collective lever) are having a negative effect on the predictions. There could be

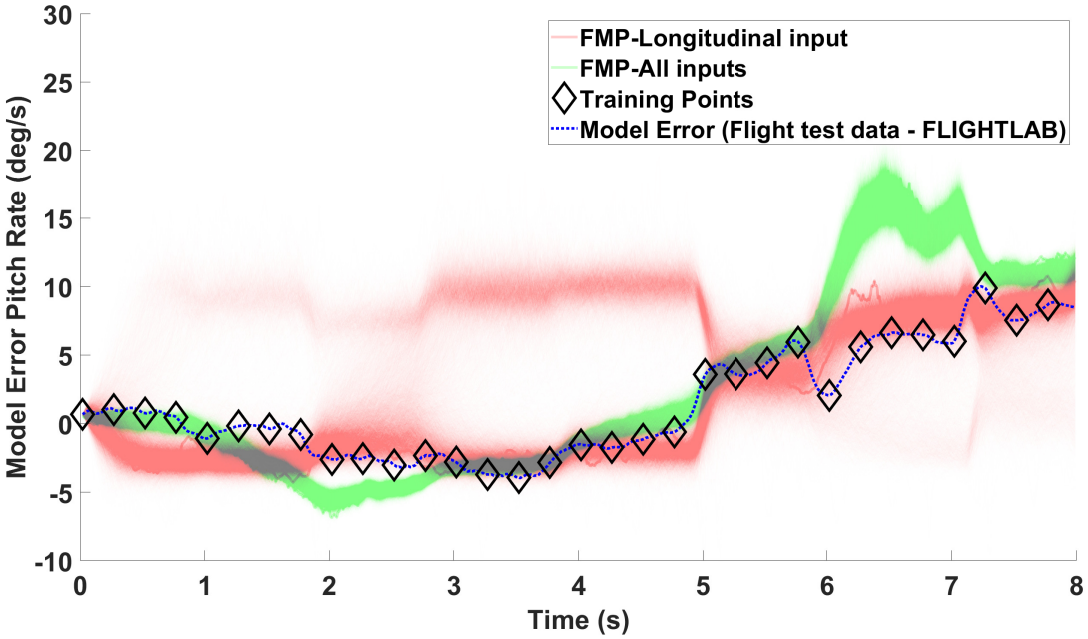


Figure 9.7: Comparison of the GP FMP realisations using different input structures (cases 2 and 4, Table 9.3) to predict the model error pitch rate.

several reasons for the result. One could be due to only one hyperparameter being used to govern the input structure; each of the inputs in the input structure would have an equal effect on the predictions as they are utilising the same hyperparameter. Therefore, the inputs that do not have a significant effect on the predictions could be affecting the predictions more than required and having a negative impact, vice versa the relevant inputs will be having a smaller effect. For example, the relevant input, in this case, could be the lagged model error pitch rate, and in the larger input structure, it will be having a reduced effect on the predictions compared to the input structure containing the longitudinal input and one lagged model error term. Another reason could be due to the GP attempting to infer a relationship between the pilot inputs and model error. If the relationship between the other pilot inputs and the model error is weak, it would have a negative effect on the GP prediction. The negative effect of the other pilot inputs in the input structure could be solved using a hyperparameter to govern each input of the input structure. If the relationship between the pilot inputs and the model error is weak, it is feasible to assume that the lagged model error term could be the most relevant input.

The similarity measure results for the grey-box model (flight test data - FLIGHTLAB) are

shown in Table 9.4, where the difference between the mean of the FMP realisations of the grey-box model and the flight test data is used to calculate the similarity measure. The input/hyperparameter structure containing the longitudinal input with one lagged model error pitch rate term utilising one hyperparameter produces the most accurate results.

Similarity measure for the comparison of the grey-box pitch rate predictions					
Prediction	**Pilot input**	**Lagged output**	**Case**	**GP Prediction**	**Similarity measure**
Pitch Rate	Longitudinal input	One lagged model error pitch rate term	1	OSAP	**0.03**
				FMP	**14.00**
		Two lagged model error pitch rate terms	2	OSAP	**0.01**
				FMP	**26.65**
	All of the current pilot inputs	One lagged model error pitch rate term	3	OSAP	**0.03**
				FMP	**112.99**
		Two lagged model error pitch rate terms	4	OSAP	**0.03**
				FMP	**55.49**

Table 9.4: The similarity measure for the model error pitch rate GP models, shown in Section 9.4.1.

Figure 9.8 displays the GP FMP realisations of the pitch rate response for the grey-box model (green realisations) and the full GP (red realisations). The grey-box model utilises the input structure containing the longitudinal input and one lagged model error pitch rate term. Note the full GP is shown in Chapter 5 and predicts the pitch rate response trained on 32 points, using the input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising one hyperparameter. Observing the comparison between the FMP realisations for the grey-box model and the full GP, it is seen that the full GP produces more accurate predictions than the grey-box model. Quantifying the error in terms of similarity measure between the mean FMP realisation and the flight test data, the grey-box model is 14.00, compared to the full GP attempting to predict the pitch rate directly, 5.62. The grey-box model is significantly worse. However, it should be noted that this method still produces better predictions than FLIGHTLAB alone.

One reason why the grey-box model is not as accurate as the full GP is that the grey-box model is attempting to predict the error between the FLIGHTLAB predictions and the flight test data using the longitudinal stick position. The correlation between the model error and the longitudinal stick position may not be as strong as the pilot inputs to the pitch rate. The relationship

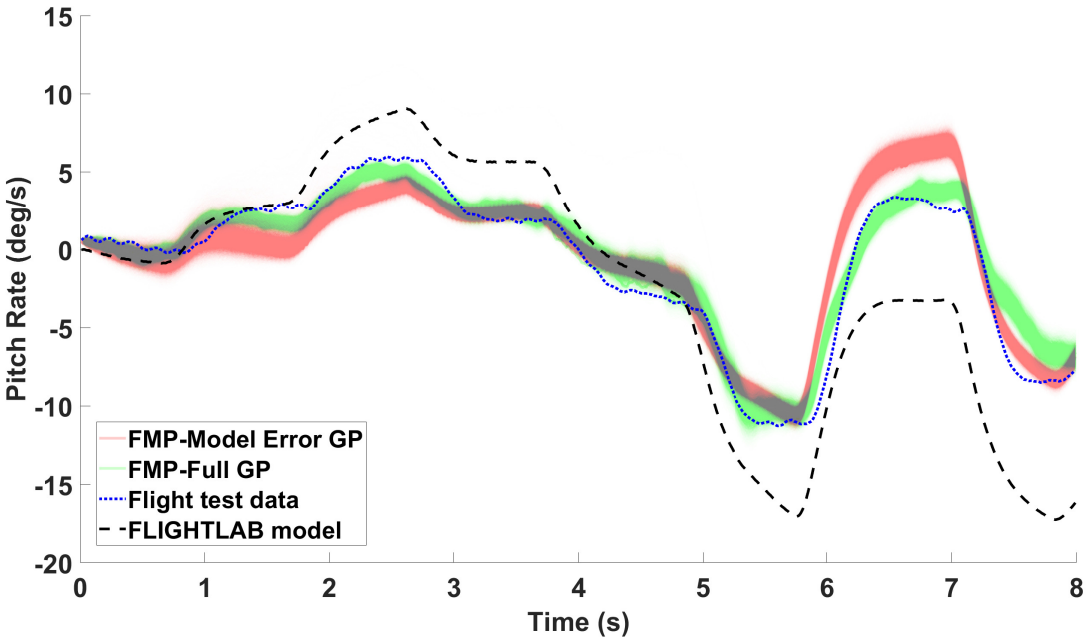


Figure 9.8: The grey-box predictions and the full GP for the pitch rate response, with a comparison to the flight test data and the FLIGHTLAB model.

of the model error to the longitudinal stick position is dependent on the FLIGHTLAB model, as the model error is calculated by the flight test data minus the FLIGHTLAB model. If the FLIGHTLAB model is not displaying a consistent relationship between its predictions and the pilot inputs, this will have a negative impact on the GP predictions, as the GP will not be able to establish a consistent relationship. If the FLIGHTLAB model is not producing a consistent relationship, this could be due to the simplifications made to the physical law based equations or it could be due to the 'tuning' of the FLIGHTLAB model. Note that given sufficient time and resources a high-fidelity FLIGHTLAB model could be created, such as those reported in [75, 76], this could improve the FLIGHTLAB model.

9.4.2 Roll Rate

The model error between the FLIGHTLAB model and the flight test data roll rate response is shown in Figure 9.9. The model error (red line) indicates the difference between the FLIGHTLAB predictions and the flight test data; this is used as the response that the GP attempts to predict. The GP predictions of model error would be then used to create a grey-box model.

Figure 9.10 displays the model error roll rate GP FMP realisations for the ARX input structures

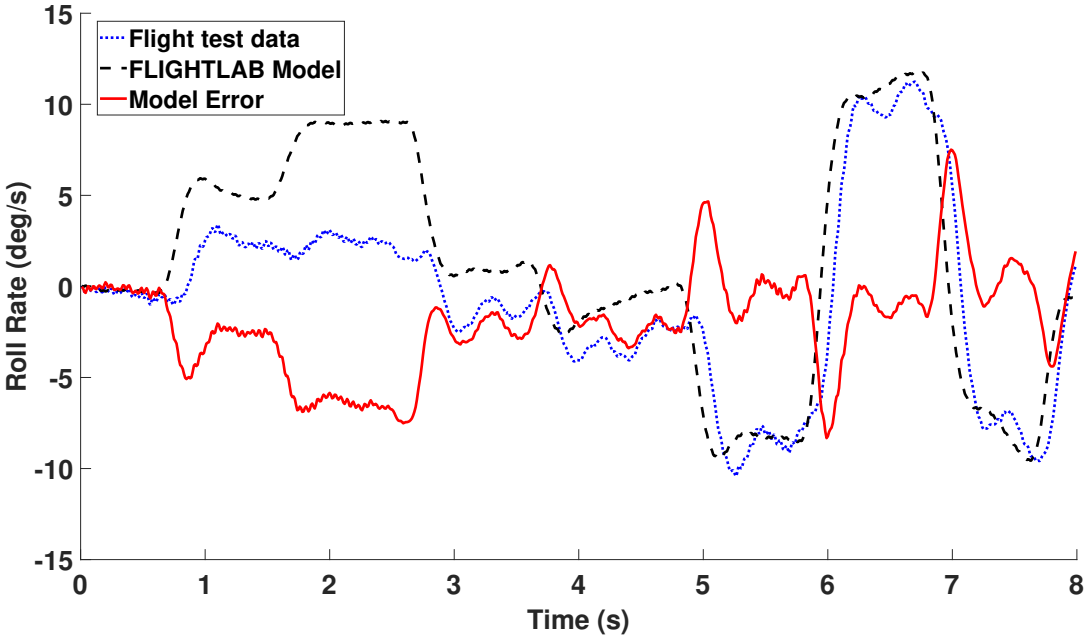


Figure 9.9: Model Error for the roll rate response (Flight test data - FLIGHTLAB model).

containing the current lateral pilot input with one lagged model error roll rate term (case 5, Table 9.3) and all of the current pilot inputs with one lagged model error roll rate term (case 7, Table 9.3). The two GP models are trained on 32 training points selected to be every 25th training point of the full dataset. Case 5 is displayed by red realisations, while case 7 is given by green realisations.

Two sets of FMP realisations are displayed in Figure 9.11, green realisations are created from the GP model utilising the input structure containing the current lateral pilot input with two lagged model error roll rate terms (case 6, Table 9.3), while red realisations are from the GP model utilising the input structure containing all of the current pilot inputs with two lagged model error roll rate terms (case 8, Table 9.3).

As can be seen from the similarity measure in Table 9.5 and Figures 9.10 and 9.11, the ARX input structure containing two lagged model error roll rate terms produces more accurate results than the GP models utilising the other ARX input structures contained one lagged model error roll rate term (cases 5, and 7). The input structure containing the lateral pilot input with two lagged model error roll rate terms produces marginally better results than the input structure containing all of the current pilot inputs with two lagged model error roll rate terms. The result

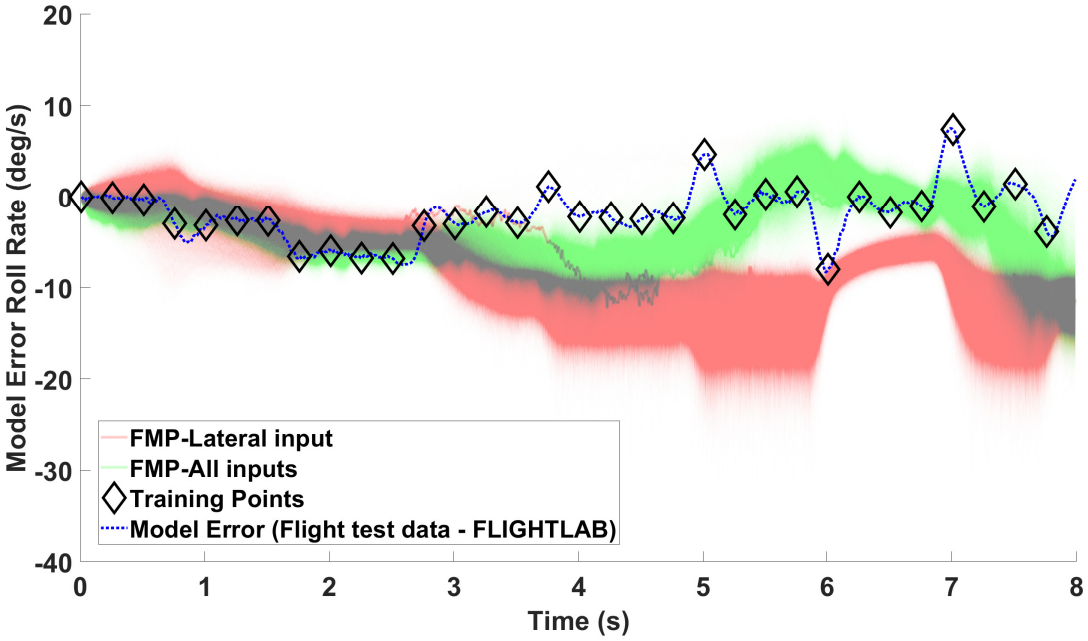


Figure 9.10: Comparison of the GP FMP realisations using different input structures (cases 5 and 7, Table 9.3) to predict the model error roll rate.

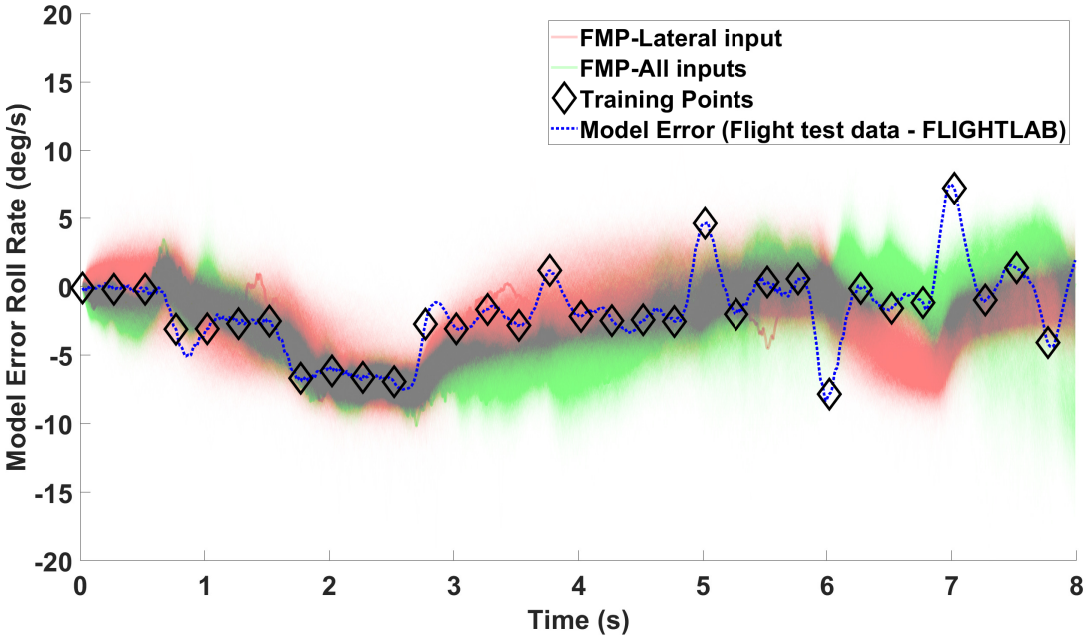


Figure 9.11: Comparison of the GP FMP realisations using different input structures (cases 6 and 8, Table 9.3) to predict the model error roll rate.

shown for the model error roll rate is similar to the model error pitch rate in that using all of the current input does not produce a more accurate model, the possible reasons for this have already been discussed. The most accurate GP model for predicting the roll rate model error uses two lagged output terms; this differs from the model error pitch rate (where one lagged output term produces the most accurate prediction); however, it is same as the GP model predicting roll rate shown in Section 5.1.2, which produces the most accurate predictions when two lagged output terms are utilised.

Similarity measure for the comparison of the model error roll rate predictions					
Prediction	**Pilot input**	**Lagged output**	**Case**	**GP Prediction**	**Similarity measure**
Roll Rate	Lateral input	One lagged model error roll rate term	5	OSAP	**0.22**
				FMP	**258.14**
		Two lagged model error roll rate terms	6	OSAP	**0.11**
				FMP	**25.69**
	All of the current pilot inputs	One lagged model error roll rate term	7	OSAP	**0.22**
				FMP	**58.22**
		Two lagged model error roll rate terms	8	OSAP	**0.18**
				FMP	**27.37**

Table 9.5: Similarity measure for the model error roll rate GP models, shown in Section 9.4.2.

The roll rate grey-box model was created with the GP predicting the model error roll rate with the input structure containing the lateral pilot input with two lagged model error roll rate terms. Figure 9.12 compares the roll rate grey-box FMP realisations to the full GP (shown in Chapter 5). The grey-box FMP realisations are given by red realisations, and the full GP is displayed by green realisations. The full GP was created using the input structure containing all of the current pilot inputs and two lagged roll rates. Observing Figure 9.12 the full GP produces more accurate FMP realisations than the grey-box model. The difference between the two GP models is confirmed by the similarity measure of 10.30 for the mean FMP of the full GP and 25.69 for the mean FMP of the grey-box. In some time locations, the FLIGHTLAB model is more accurate than the grey-box realisations, such as time to 7 to 8 seconds. The inaccuracy of the grey-box model is likely to be the same reason as previously for the model error pitch rate. The GP is attempting to predict the difference between the FLIGHTLAB model and the flight test data; the inputs used may not have any strong correlation with the roll rate model error.

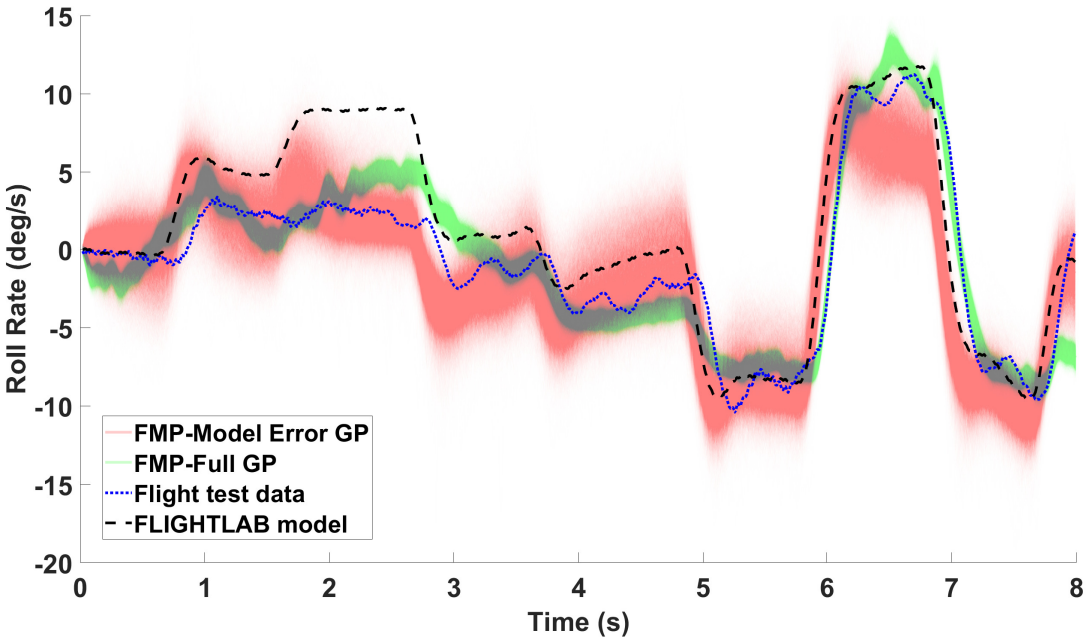


Figure 9.12: The grey-box predictions and the full GP for the roll rate response, with a comparison to the flight test data and the FLIGHTLAB model.

9.4.3 Yaw Rate

Figure 9.13 displays the model error between the flight test data and the FLIGHTLAB predictions for the yaw rate response. The yaw rate model error (red line) is used to create the GPs shown in the current section. It is clear from observing Figure 9.13 that towards the end of the time series, the FLIGHTLAB model does not capture the correct dynamics, and predicts the opposite way to that of the flight test data. The FLIGHTLAB predictions could be due to the simplifications to the physical law based equation or could be due to the 'tuning' of the model.

Figure 9.14 displays two sets of GP FMP realisations predicting the model error yaw rate, both trained on 32 training points selected every 25th point of the full data set. The difference between the two sets is the ARX input structure. The red realisations display the model error GP FMP realisations for the input structure containing the current pedal pilot input with one lagged model error yaw rate term (case 9, Table 9.3). The model error GP FMP realisations for the input structure containing all of the current pilot inputs with one lagged model error yaw rate term (case 11, Table 9.3) is shown by green realisations.

Two sets of FMP realisations are displayed in Figure 9.15. Red realisations show the model

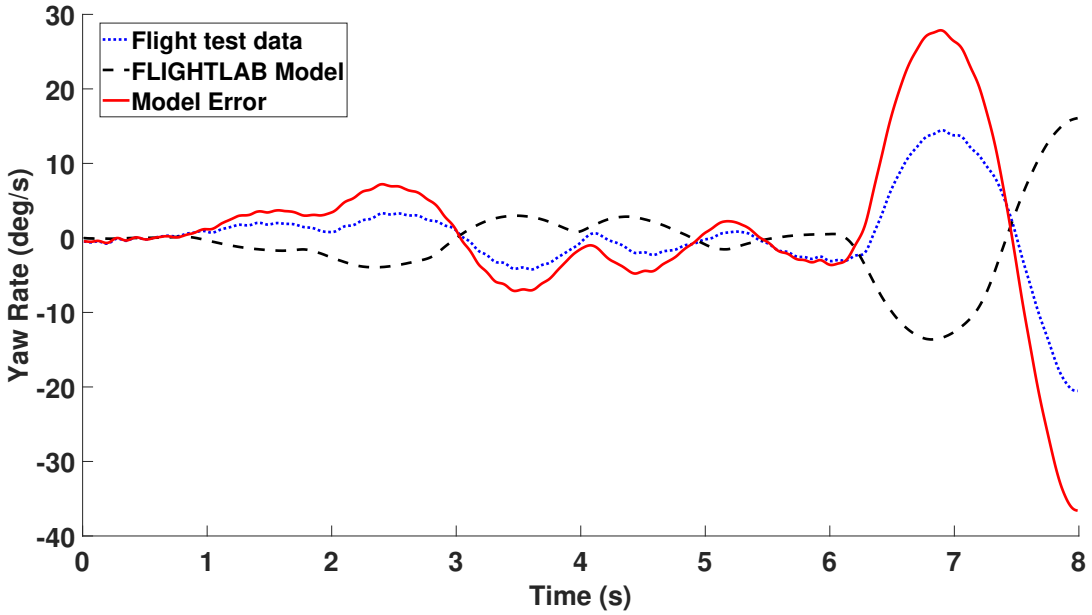


Figure 9.13: Model Error for the yaw rate response (Flight test data - FLIGHTLAB model).

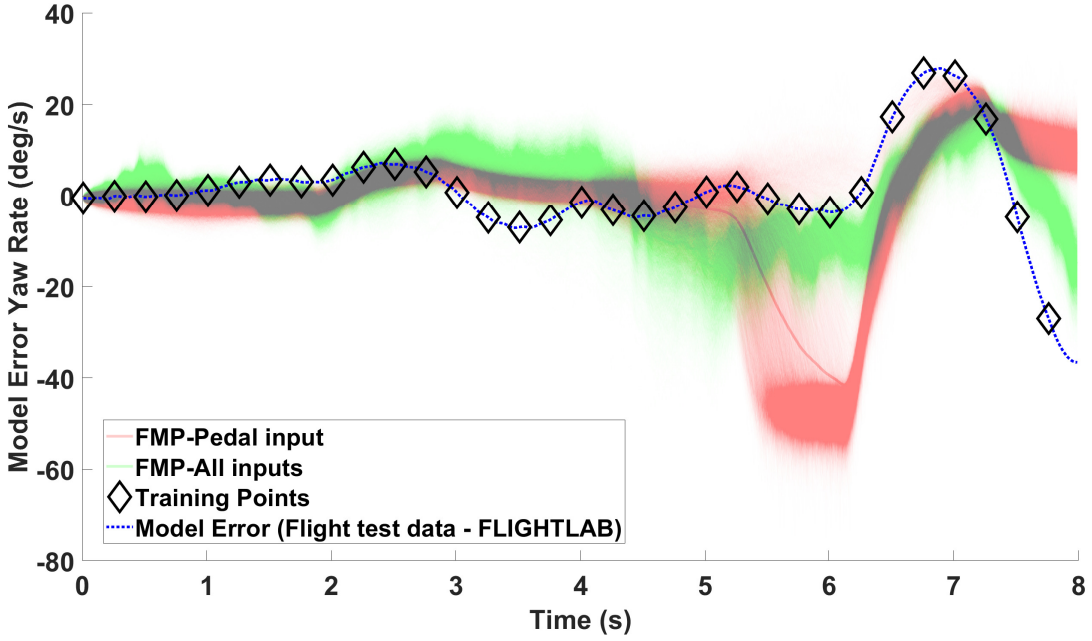


Figure 9.14: Comparison of the GP FMP realisations using different input structures (cases 9 and 11, Table 9.3) to predict model error yaw rate.

error FMP realisation for the input structure containing the current pedal pilot input with two model error yaw rate terms (case 10, Table 9.3), and the green realisations are from the model error FMP realisations for the input structure containing all of the current pilot inputs with two lagged model error yaw rate terms. Both models are trained on 32 training points which are displayed as black diamonds in Figure 9.15.

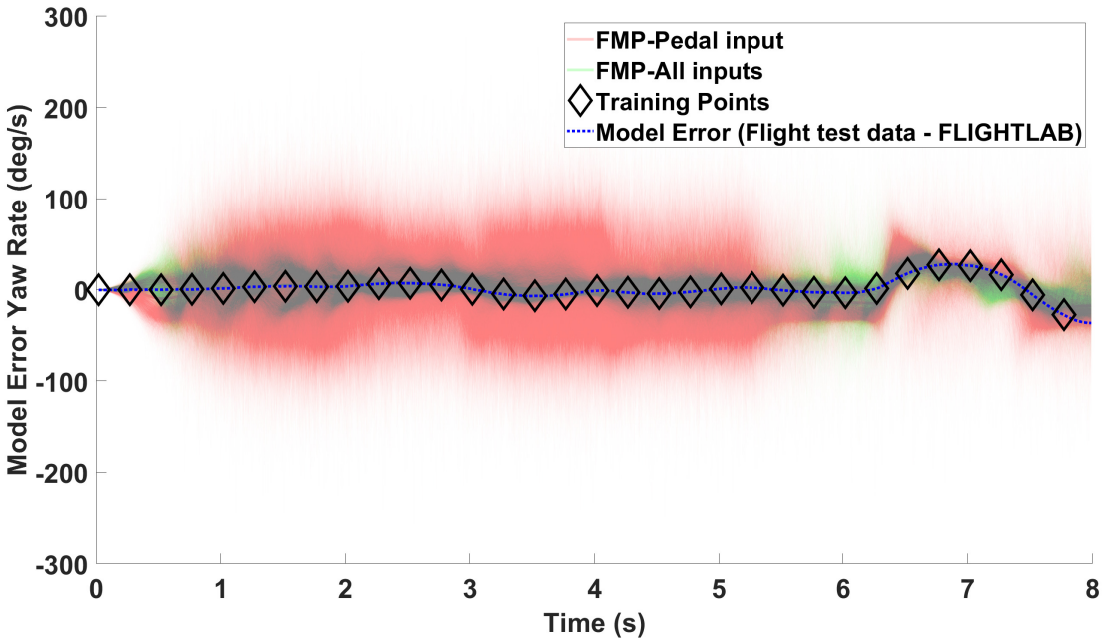


Figure 9.15: Comparison of the GP FMP realisations using different input structures (cases 10 and 12, Table 9.3) to predict model error yaw rate.

Using the similarity measure (the difference between the flight test data and the grey-box model) in Table 9.6 and Figures 9.14 and 9.15, it is evident that the input structure containing all of the current pilot inputs with two lagged model error yaw rate terms is required to produce a more accurate prediction. For the yaw rate grey-box model, the GP predicting the model error yaw rate uses an input structure containing all of the current pilot inputs with two lagged model error yaw rate terms.

As with the other sections in the current chapter, the simulated annealing and OSAP figures are given in Appendix I.5.

Figure 9.16 displays the grey-box FMP realisations, which are shown as red realisations. The green realisations display the corresponding GP FMP realisations for the full GP (Chapter 5). The full GP was created utilising the input structure containing all of the current pilot inputs

Similarity measure for the comparison of the model error yaw rate predictions					
Prediction	**Pilot input**	**Lagged output**	**Case**	**GP Prediction**	**Similarity measure**
Yaw Rate	Pedal input	One lagged model error yaw rate term	9	OSAP	**0.15**
				FMP	**725.89**
		Two lagged model error yaw rate terms	10	OSAP	**0.01**
				FMP	**216.10**
	All of the current pilot inputs	One lagged model error yaw rate term	11	OSAP	**0.22**
				FMP	**321.99**
		Two lagged model error yaw rate terms	12	OSAP	**0.02**
				FMP	**137.04**

Table 9.6: Similarity measure for the model error yaw rate GP models, shown in Section 9.4.3.

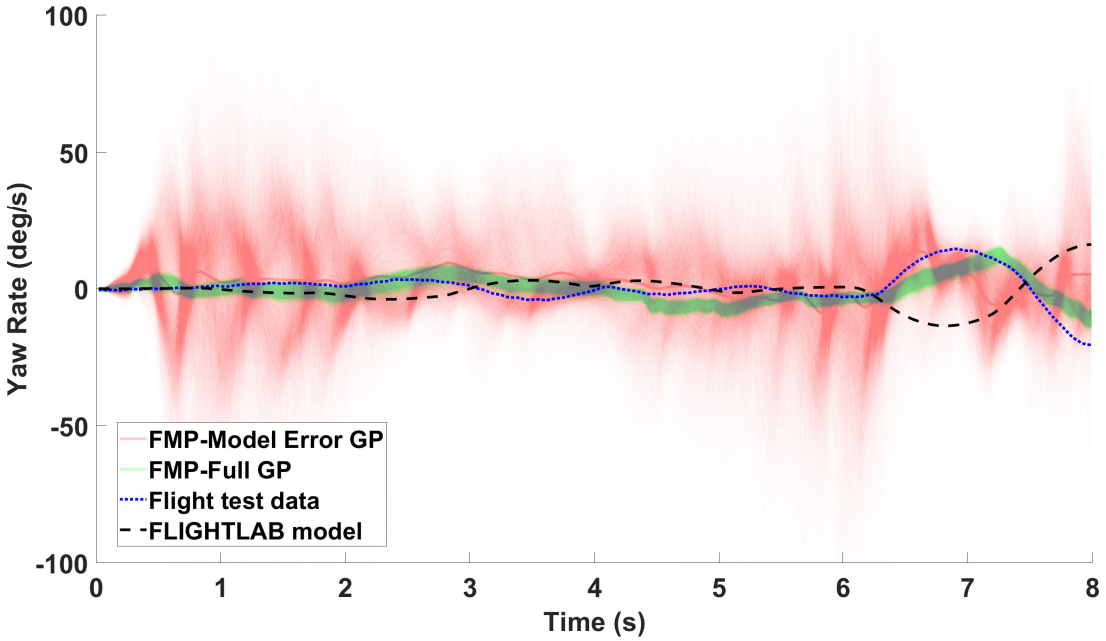


Figure 9.16: The grey-box predictions and the full GP for the yaw rate response, with a comparison to the flight test data and the FLIGHTLAB model.

and two lagged yaw rate terms. The full GP produces more accurate FMP realisations compared to the grey-box model. The grey-box FMP realisations struggle to capture the dynamics of the yaw rate response; however, this is also the case for FLIGHTLAB. Using Figure 9.16, it is not clear if the grey-box or FLIGHTLAB model creates the most accurate predictions compared to the flight test data. The similarity measure for the FLIGHTLAB model is 370.08 and for the mean grey-box FMP realisations, 137.04. The results from Table 9.6 and Figure 9.16 provides evidence that the grey-box approach can create a more accurate model regarding the yaw rate response than the FLIGHTLAB model. The grey-box models are not as accurate as predicting the dynamics directly; this could be due to what has already been discussed that the FLIGHTLAB model is not predicting a consistent relationship between the model error and the pilot inputs.

9.4.4 Heave

Figure 9.17 displays the model error between the heave flight test data response and the corresponding FLIGHTLAB model. The GPs created in this section use the model error response displayed by the red line in Figure 9.17.

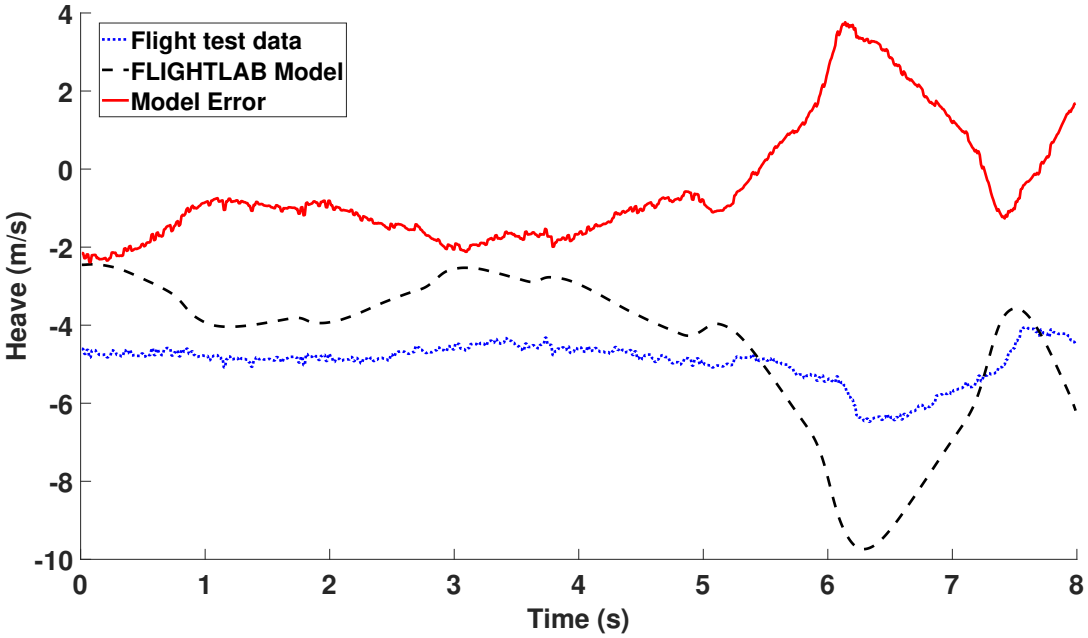


Figure 9.17: Model Error for the yaw rate response (Flight test data - FLIGHTLAB model).

Figure 9.18 displays two sets of model error heave GP FMP realisations; the difference be-

tween the two sets is the ARX input structure that has been used. The red realisations are for the model error heave GP FMP realisations using the input structure containing the current collective pilot input with one lagged model error heave term (case 13, Table 9.3). The green realisations display the model error heave GP FMP realisation using the input structure containing all the pilot inputs with one lagged model error heave term (case 15, Table 9.3).

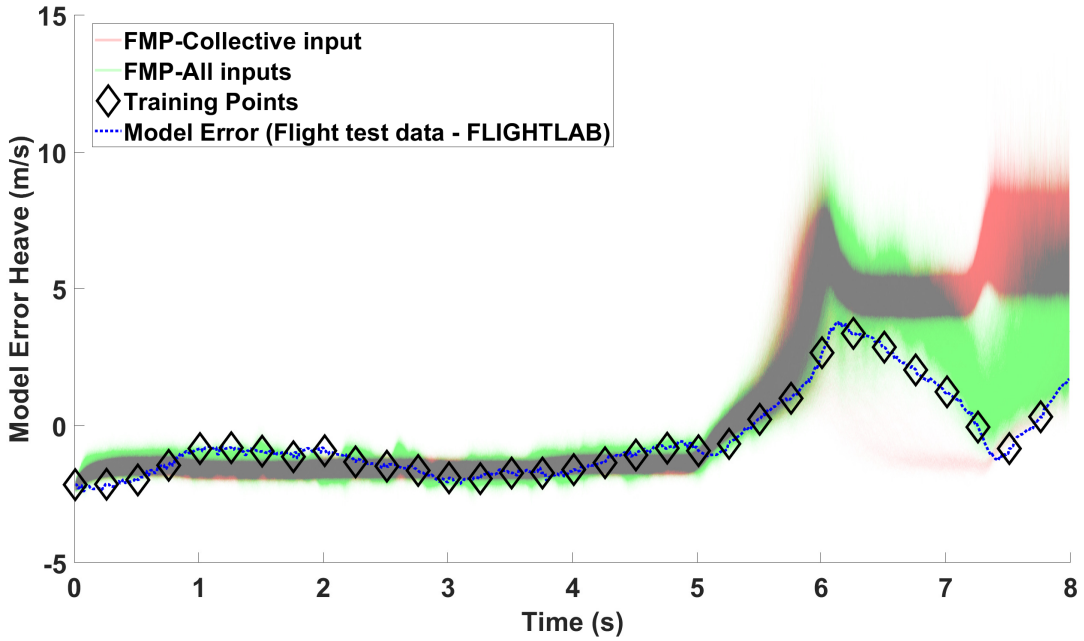


Figure 9.18: Comparison of the GP FMP realisations using different input structures (cases 13 and 15, Table 9.3) to predict model error heave.

Two sets of GP FMP realisations are displayed in Figure 9.19, the red realisations are from the GP model utilising the input structure containing the collective pilot input and two lagged model error heave terms (case 14, Table 9.3) and the green realisations are from the GP model utilising the input structure containing all of the current pilot inputs and two lagged model error heave terms (case 16, Table 9.3).

Using Figures 9.18 and 9.19, it is difficult to judge which input structure creates the most accurate realisations. The similarity measure between the flight test data and the grey-box (FLIGHTLAB + model error GP) FMP realisations is given in Table 9.7. The similarity measures confirm that the input structure containing all of the current pilot inputs with one lagged model error heave term (case 15, Table 9.3) and the current collective pilot input with two lagged model error heave terms (case 14, Table 9.3) creates the more accurate FMP realisations.

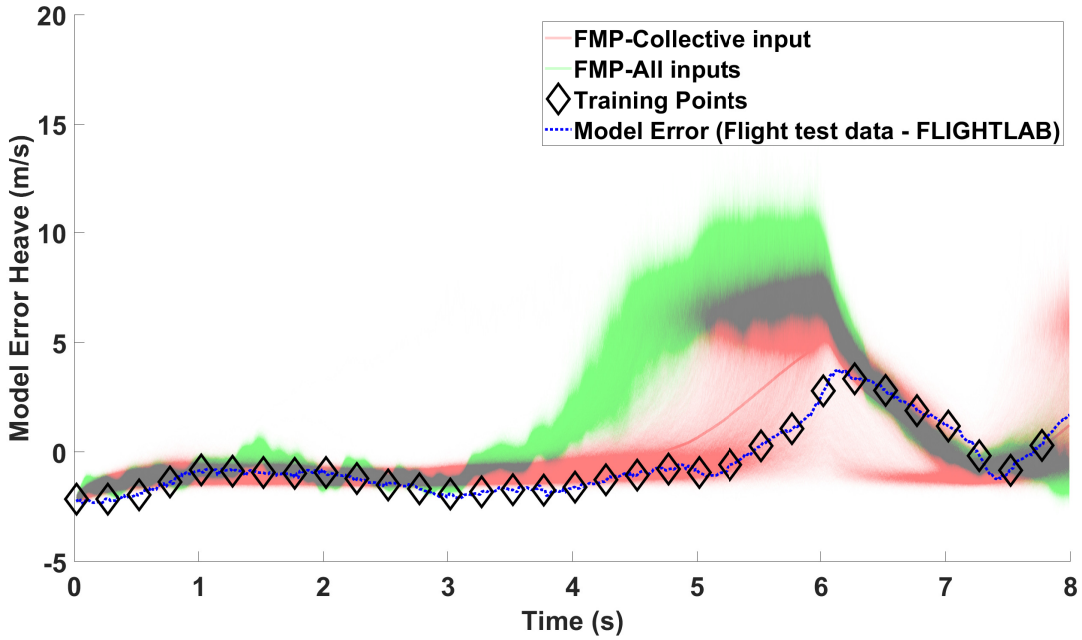


Figure 9.19: Comparison of the GP FMP realisations using different input structures (cases 14 and 16, Table 9.3) to predict model error heave.

Similarity measure for the comparison of the model error heave predictions					
Prediction	**Pilot input**	**Lagged output**	**Case**	**GP Prediction**	**Similarity measure**
Heave	Collective input	One lagged model error heave term	13	OSAP	**1.75**
				FMP	**1868.72**
		Two lagged model error heave terms	14	OSAP	**1.93**
				FMP	**466.66**
	All of the current pilot inputs	One lagged model error heave term	15	OSAP	**1.70**
				FMP	**466.84**
		Two lagged model error heave terms	16	OSAP	**2.61**
				FMP	**4743.54**

Table 9.7: Similarity measure for the model error heave GP models, shown in Section 9.4.4.

For the grey-box model, the GP predicting the model error heave uses the input structure containing all of the current pilot inputs with one lagged model error heave term. Note, the GP utilising the input structure of the current collective pilot input and two lagged model error terms produced similar results in terms of the similarity measure, therefore, either could be selected.

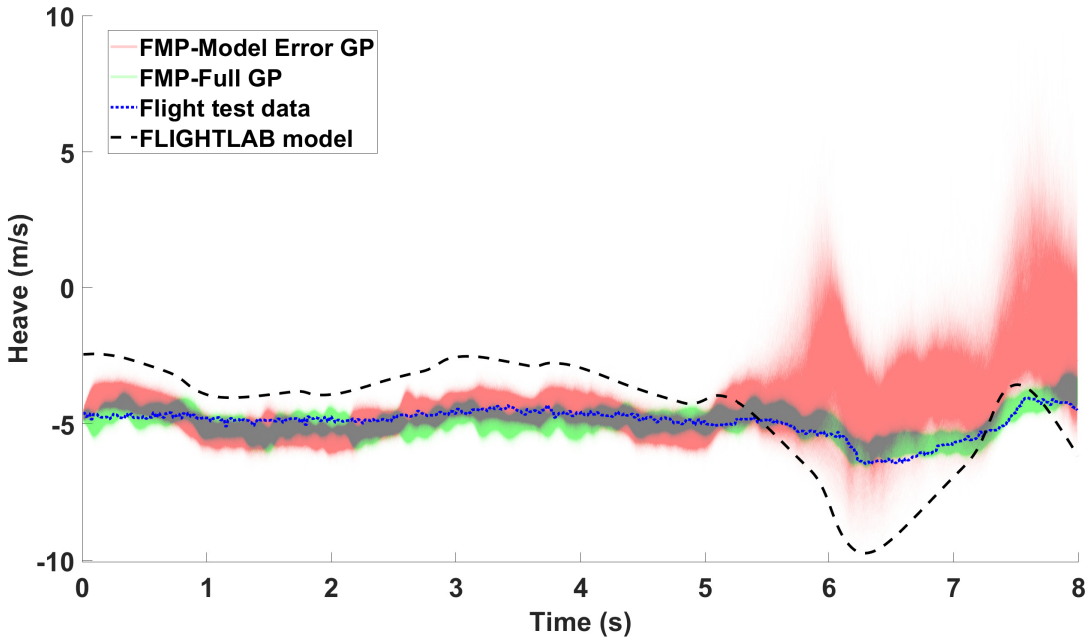


Figure 9.20: The grey-box predictions and the full GP for the heave response, with a comparison to the flight test data and the FLIGHTLAB model.

Figure 9.20 compares the grey-box model to the full GP as well as the corresponding FLIGHTLAB model. The full GP was created using an input structure containing all of the current pilot inputs with one lagged heave term. Visually using Figure 9.20 the full GP provides better FMP realisations than that of the grey-box model, this could be due what has already been previously discussed, that the correlation between the pilot inputs and the model error heave may not be significant. If this is the case, the GP will struggle to capture the dynamics of the model error heave. The grey-box FMP realisations manage to capture the dynamics of the heave response for the first part of the time series (up to time 5 seconds), however, as the time series progresses the GP model degrades. Comparing the grey-box model to the FLIGHTLAB model, the grey-box model outperforms the FLIGHTLAB model. Quantifying the difference in terms of the similarity measure between the prediction and the flight test data, the FLIGHT-

LAB model has a similarity measure of 956.60 and the mean FMP realisations for the grey-box model, 466.84, thus confirming that the grey-box model produces more accurate predictions than the FLIGHTLAB model.

The simulated annealing and OSAP figures for the GP shown in this section are also displayed in Appendix I.6.

9.5 Summary

Firstly, note that the full GP outperforms the corresponding grey-box model for each response. The grey-box model is attempting to predict the difference between the flight test data and the FLIGHTLAB model. The FLIGHTLAB model should capture the dynamics of the response and, therefore, one might expect it to be easier for the GP to establish a relationship between the pilot inputs and the flight test data, compared to when the GP is attempting to find the relationship without any prior knowledge of the dynamics. However, the grey-box models performed worse than the full GP; this could be due to what has been discussed previously, the grey-box model is attempting to predict the response of the model error; therefore it is using the correlation between the pilot inputs and the error between the flight test data and the FLIGHTLAB model. If the FLIGHTLAB predictions were not consistent, the GP would struggle to establish a relationship between the pilot inputs and the model error. Note, as was previously mentioned, if the FLIGHTLAB model was already 'tuned', given more time effort, the FLIGHTLAB model could have been created to make more accurate predictions. The FLIGHTLAB model is causing the poor results in the model error pitch rate prediction when the input structure containing all of the current pilot inputs is utilised, as it is known that the GP can establish a relationship between all of the current pilot inputs and the flight test data, as shown in Chapter 5. As previously discussed the inaccuracy of the model error pitch rate using the ARX input structure containing all of the current pilot inputs, could be due to a combination of the high relevance of the lagged model error pitch rate term and the use of one hyperparameter to govern the GP input structure.

Table 9.8 displays the similarity measure between the flight test data and the mean FMP realisations for the models displayed in the current chapter with comparison to the FLIGHT-

LAB and full GP models. The table enables easy comparison between the GP models and the FLIGHTLAB models. The sparse GP models were not created for the model error, as it was clear that the GPs were struggling to capture the dynamics of the model error.

What is clear from the results displayed in the current chapter and from Table 9.8 is that with the current datasets, it is more beneficial to use a black-box model to make predictions. However, given a FLIGHTLAB model that had been 'tuned' in another manner, the results may have been different, and the grey-box models may have provided better results.

Similarity measure for grey-box FMP realisations comparison to full GP and FLIGHTLAB			
Manoeuvre	**Prediction**	**Model**	**Similarity measure**
3-2-1-1 Longitudinal Input	Pitch Rate	Full GP (32 Points)	**5.62**
		Grey-box model (32 Points)	**14.00**
		FLIGHTLAB Model	**75.02**
3-2-1-1 Lateral Input	Roll Rate	Full GP (32 Points)	**10.30**
		Grey-box model (32 Points)	**25.69**
		FLIGHTLAB Model	**42.43**
3-2-1-1 Pedal Input	Yaw Rate	Full GP (32 Points)	**53.26**
		Grey-box model (32 Points)	**137.04**
		FLIGHTLAB Model	**370.08**
3-2-1-1 Collective Input	Heave	Full GP 32 Points	**34.45**
		Grey-box model (32 Points)	**466.84**
		FLIGHTLAB Model	**956.60**

Table 9.8: Similarity measure for grey-box model with comparison to the full GP, Bo105 flight test data and FLIGHTLAB models.

Chapter 10

Conclusion and Discussion

10.1 Conclusions

Gaussian process (GP) machine learning techniques have been used to develop rotorcraft response predictive models. The performance of these models has been assessed by comparing the accelerations and velocities (or 'rates') of the predictions to the acceleration and velocities of the flight test data. The model predictions are also compared to a corresponding physics based model (FLIGHTLAB). From this investigation, the following conclusions can be drawn.

The GP model using an ARX input structure produced predictions that were comparable to the flight test data and outperformed the FLIGHTLAB model.

The feasibility study in Chapter 4 concluded that the GP model was more accurate for the on-axis pitch rate response than a linear regression model. The most accurate input structures were the ARX and ARMA; however, it was chosen to use the ARX input structure as this is consistent with other literature (see studies by Kumar et al. [43] and Omkar et al. [44]). The GP model utilising an ARX input structure containing the longitudinal stick position with one lagged response term was able to predict the on-axis pitch rate and the off-axis roll rate more accurately than the existing FLIGHTLAB model. The GP model for the on-axis response had a higher fidelity than the off-axis response; however, this was expected, as stated by Padfield that the off-axis motion is unpredictable [3]. Accordingly, it was decided that the on-axis response would be investigated in the thesis.

Using an ARX input structure that contained only the primary on-axis pilot input with one lagged on-axis response term for that axis was found to give adequate response predictions. However, the most appropriate input structure for the proposed machine learning approach, i.e. the one that gave the most accurate results in terms of flight test response and accelerations, was found to be one where all of the current pilot inputs with two lagged on-axis response terms were used. This was found to be true for the pitch rate, roll rate and yaw rate. For the heave model, the input structure containing all of the current pilot inputs and one lagged response term produced the most accurate results for the response and accelerations. The GP model predicting the heave response produced promising results but were inconsistent between the different input structures. For example, using the input structure containing all of the current pilot inputs and two lagged response terms produce worse predictions than the input structures utilising one and three lagged response terms, which would not be expected as the second lag term is associated with the acceleration of the response. With the other responses (pitch rate, roll rate and yaw rate) the input structure containing two lagged response terms produced the most accurate predictions. It is worth noting, that the FLIGHTLAB model also struggled to predict the heave response. Different hyperparameter structures were investigated to see if utilising a hyperparameter for each input of the GP input structure produces more accurate predictions than using one hyperparameter to govern the whole input structure. It was concluded that the GPs using one hyperparameter produce more accurate predictions. It was also found that the hyperparameter uncertainty did not have a large effect on the GP predictions and, as a result, was not considered further in the remainder of the thesis.

To be able to create a GP for a full flight test campaign, it is essential to reduce the number of training points, to reduce the cost associated with the training phase. The variational sparse GP approach used 32 training points for each response and a reduced training set for the pitch rate, roll rate and yaw rate and heave. These sparse models generally performed almost as well as the full GP, for the pitch rate, roll rate and heave and still generally gave more accurate results than the physics based model that was being investigated. For the yaw rate, the sparse GP predictions were not as accurate as the full GP, which could be due to the random selection of the greedy optimisation. This technique, therefore, shows promise for use in real-world

settings where the volume of available training data will, potentially, be significantly larger. Regarding the real-time running capabilities of the GP model, for the sparse GP models using 10 training points a full set of realisations (8 seconds of predictions) takes between 0.03-0.045 seconds and for the sparse GP model using 15 training points, it takes between 0.035 - 0.05 seconds. The time to make predictions provides evidence that the GP can predict sufficiently quickly for flight simulation purposes.

10.2 Discussion

In an attempt to assess the utility of GP machine learning modelling techniques for rotorcraft flight dynamics real-time response predictions, several different GP models have been developed using flight test data as the basis. The use of the GP techniques allows a statistical analysis of the response to be performed to provide confidence levels in the predictions made.

Chapter 4 assessed the feasibility of using machine learning modelling techniques to predict the on-axis response of the Bo105 rotorcraft. The GP models produced more accurate results than the linear regression models; therefore, the use of the GP model was the focus for the remained of the thesis. The 'one step ahead' models that were shown in Chapter 4 were able to predict the on-axis (pitch rate) and off-axis (roll rate) response very accurately and with high confidence. Of course, such models are of little/no use for flight simulation purposes as they are only able to make a prediction one time-step into the future. So, while the results were impressive, their utility is limited in the context of the current work. For the on-axis pitch rate response, the GP model using an ARX input structure created GP FMP that were accurate and outperformed the FLIGHTLAB model. The FMP are a more stringent test of the GP model. From the results of the feasibility Chapter 4, it was concluded that using the GP model using an ARX input structure for the on-axis responses was the most viable option.

Response predictions were then made for the on-axis response, in Chapter 5, using two sets of ARX input structures each with three different input structures, the first containing the primary control input with one, two and three lagged on-axis response terms, and the second consisted of all of the current pilot inputs with one, two and three lagged on-axis response terms. The ARX input structures were able to make subjectively good predictions of the rotorcraft

response. The ARX input structure containing all of the current pilot inputs with two lagged response terms created the most accurate models for the pitch rate, roll rate and yaw rate.

It was also noted that the FMP did not always encompass all of the training data; this could be due to not using the other pilot inputs in the input structure. The other pilot inputs were included in Chapter 5 and the resulting FMP did not always encompass the training data, therefore, the FMP not encompassing the training data is not caused by the exclusion of the other pilot input. The FMP not encompassing all of the training data could be due to the FMP approach, in which the uncertainty is propagated. The previous (uncertain) predictions, therefore, become inputs and carry their predictive uncertainty into future predictions [40]. Towards the latter stages of prediction, the GP model has made many predictions from uncertain inputs and can 'drift' from the true response, therefore the predictions may not encompass all of the training data.

The GP model can utilise one hyperparameter or a hyperparameter for each input of the GP input structure. Chapter 6, investigated which hyperparameter structure produce the most accurate predictions, using the same on-axis responses (pitch rate, roll rate, yaw rate and heave) as shown in Chapter 5. The ARX input structure containing all of the current pilot inputs with two lagged on-axis response terms was used for the pitch rate, roll rate and yaw rate in Chapter 5. For the heave, the input structure utilising all of the current pilot inputs with one lagged heave term was used. It was noted that the optimisation technique might have struggled to find the optimal hyperparameters in the 6/7-dimensional space.

Given the limited amount of data used to train (32 training points) the models in Chapters 5 and 6, their subsequent performance is notable. The interesting point about the results, compared to a traditional flight dynamics approach, is that the prediction responses are provided with the confidence level with which the model has been able to make that prediction. The GP using 32 training points (selected to be every 25th point) with the ARX input structure consisting of all of the current pilot inputs with two lagged on-axis response terms utilising one hyperparameter was referred to as the full GP for the pitch rate, roll rate and yaw rate. The full GP for the heave was created using the input structure containing all of the current pilot inputs with one lagged response term.

The variational sparse GP method firstly attempted to use the same number of training points as the full GP to see if it could make a comparable model. This step was used for code verification and to investigate if the method of greedy selection created a comparable model to that of the full GP. The use of a reduced training set for the sparse GP method was then investigated. The full GP used 32 training points, evenly spaced over the time series (every 25th data point), while the variational sparse GP using a reduced training set utilised 12 training points for the pitch rate, 20 training points for the roll rate, 15 training points for yaw rate, and the heave response utilised 15 training points. The results are shown in Section 8.2 provide evidence that using a reduced training set can be used to produce a similar model. Using a reduced number of training points would be beneficial in terms of scalability, allowing this method to be applied to larger data sets. The reduced training set models provided more accurate predictions than the equivalent FLIGHTLAB model.

There are known issues with machine learning techniques. Specifically, when used correctly, they are very good at interpolating within the training data but much less so at extrapolating beyond it. Similarly, they are 'black-box' techniques that utilise model states that are difficult or impossible to interpret in a physical sense. To the first point, there are much larger datasets available for production aircraft, and so the available flight envelope of data in which to interpolate should cover the normal operational environment. It is acknowledged that these techniques will likely not cope well in flight regimes that have not been encountered in the training data. However, the statistical nature of the methods presented may provide a means by which erroneous response predictions can be filtered out. To the second point, even in physical based models, depending upon the structure used, measurement data for all of the internal physical variables may not be available, reducing the accuracy and applicability of the model. This presents a conundrum as to whether to use a reduced accuracy physics based model, where all of the internal states are understood or a potentially more accurate model where they are not. As stated previously, the methods presented are not intended to replace physics based modelling, but to complement and enhance accuracy when necessary – the so-called grey-box model.

The idea of the 'grey-box' approach was investigated in Chapter 9, where the physical law

based FLIGHTLAB model (referred to as a 'white-box' model) is combined with the machine learning regression model (referred to as a 'black-box' model) predicting the error between the FLIGHTLAB model and the flight test data. What is evident is that the GP predicting the response directly produced more accurate predictions than the grey-box model. The superior accuracy of the full GP is due to the GP attempting to predict the response directly by using the correlation between the pilot inputs and the response of the rotorcraft. The model error GP is attempting to predict the error between the FLIGHTLAB model and the flight test data; this error may not have a strong correlation to the pilot inputs. If there is little correlation, the GP will struggle to prediction the model error, in which in some cases it does struggles such as the yaw rate.

In all cases, the GP model responses were compared not only with the flight test data to assess the prediction accuracy but also with a FLIGHTLAB physics based model available. The similarity measure between the prediction and the flight test data for all the models are shown in the thesis are given in Table 10.1. The FLIGHTLAB model was generally able to give good prediction accuracy over the initial part of the manoeuvres tested; the GP models were able to provide more accurate response predictions over the majority of the manoeuvre period. In that sense, the GP models 'outperformed' the FLIGHTLAB model in this study. The FLIGHTLAB model used was not intended to be a high-fidelity representation of a Bo105 aircraft but was meant to be 'representative' of it. It is recognised that higher fidelity models can be created given sufficient time and resources, such as those reported in [75, 76]. One possible use envisaged for these machine-learning modelling techniques is to complement existing physics based techniques and 'take-over' when the physical models become less accurate.

Pitch Rate – full GP provides the best predictions closely followed by the sparse GP using the same number of training points. The reason the sparse GP predictions do not match the full GP could be due to the greedy approach of selecting the training points. The sparse GP using 12 training points, produces a GP model with predictions that perform very promising. Note that all of the GP models outperform the FLIGHTLAB model. The grey-box model produces more accurate predictions than the FLIGHTLAB model, showing evidence that the grey-box approach can be used to predict the pitch rate response.

Roll Rate – the sparse GP using 20 training points produces the most accurate predictions, but the full GP also produces suitable accurate predictions. All models outperform the FLIGHTLAB model, providing evidence that the grey-box approach can be used to improve the predictions of the roll rate response.

Yaw Rate – the full GP for the yaw rate response produced the most accurate predictions, while the other methods struggled to match the performance of the full GP. The sparse GP using 32 training points, did not produce a model that was comparable to the full GP. This could be due to the random element of the sparse GP approach, that attempts to find the optimal training points. The model error GP exceeds the performance of the FLIGHTLAB model. It is, therefore, feasible to assume that the model error GP can improve the FLIGHTLAB prediction

Heave – the sparse GP for the heave response using 15 training points produces the most accurate prediction. This could be because, when using 32 training points, for the sparse GP model it created predictions that are overfitted.

The results shown in the current thesis provide evidence that the GP is able to outperform a physical law based model. The author has assessed the performance of the GP model against the equivalent physical law based model, however, quantification of the performance of the model will depend on the role the simulator is used for. Simulators are fundamental to aviation design and development, testing and qualification activities as well as in training and research [1, 2]. Depending on the role that the simulator is utilised for (rotorcraft development, qualification or training), different levels of fidelity (closeness to the real rotorcraft) would be acceptable. During the rotorcraft design and development phase, there may not be a rotorcraft created in which data can be taken from. While data can be generated by using the properties of the designed rotorcraft (from the physical law based equations), it is envisaged the GP model would not be best used in the development stage, as there is little benefit in creating a GP model trained on data generated from physical law based equations that utilise model properties. The GP model for the simulator would be best utilised during the training and research phases. Using the GP model for training and mission rehearsal, limits the range of the flight envelope

Similarity measure all the models displays in the thesis			
Manoeuvre	**Prediction**	**Model**	**Similarity measure**
3-2-1-1 Longitudinal Input	Pitch Rate	Full GP (32 Points)	**5.62**
		Sparse GP (32 Points)	**8.54**
		Sparse GP (12 Points)	**16.59**
		Grey-box model (32 Points)	**14.00**
		FLIGHTLAB Model	**75.02**
3-2-1-1 Lateral Input	Roll Rate	Full GP (32 Points)	**10.30**
		Sparse GP (32 Points)	**15.29**
		Sparse GP (20 Points)	**9.62**
		Grey-box model (32 Points)	**25.69**
		FLIGHTLAB Model	**42.43**
3-2-1-1 Pedal Input	Yaw Rate	Full GP (32 Points)	**53.26**
		Sparse GP (32 Points)	**129.45**
		Sparse GP (15 Points)	**171.06**
		Grey-box model (32 Points)	**137.04**
		FLIGHTLAB Model	**370.08**
3-2-1-1 Collective Input	Heave	Full GP 32 Points	**34.45**
		Sparse GP (32 Points)	**141.73**
		Sparse GP (15 Points)	**31.01**
		Grey-box model (32 Points)	**466.84**
		FLIGHTLAB Model	**956.60**

Table 10.1: Similarity measure for all the GP models shown in the thesis.

that the GP has to be trained on, and therefore the amount of data required to accurately predict the mission or training manoeuvres. A possible approach in future work could be to create a GP model to capture the whole flight envelope of the rotorcraft. The next logical steps are to investigate if the GP model could create accurate models for training and mission rehearsal. For example, mission rehearsal could be a task where the pilot has to fly to a specific location to perform a rescue. The 3-2-1-1 manoeuvres described in Chapter 3 are examples of 'specific manoeuvres' which are designed to excite different frequencies at a particular speed and altitude. The full flight test campaign can include multiple 'specific manoeuvres' at different parts of the flight envelope (at varying speeds and altitudes). The full flight test campaign produces a dataset, which potentially captures a large proportion of the flight envelope while exciting different frequencies, and therefore has the potential to generalise over a mission or a range of missions.

The GP has a useful property in being able to quantify the confidence in the predictions. A high level of confidence normally indicates the GP has been trained on an adequate number

of data points. It is, therefore, possible to use this feature to advise where more training data is required in specific regions, as the confidence in the prediction in the given region would be low and displayed by wider confidence bounds (see the example of the GP predicting a sine wave in Figures 7.1 and 7.2). This assessment can be undertaken during the development of the GP model and be subject to the role that the simulator is being used for. For example, if the simulator was going to be utilised for mission rehearsal, it would be envisaged that the GP would need to capture all extremes of the mission; this would be defined by the flight envelope, such as one extreme could be while the missions is at low speed and altitude. The use of the confidence levels would advise if the GP requires more data in specific regions. However at the current stage it is difficult to judge the size of the datasets required for the GP to capture the dynamics of the rotorcraft for simulation purposes; this assessment would have to be undertaken during the development stage of the GP model. It is envisaged that the size of the dataset will be driven by the complexity and the duration of the mission rehearsal of interest. It is anticipated that using the confidence levels to inform where the GP requires more training data will create a more accurate model in terms of velocity and acceleration of the rotorcraft. The approach could also provide an automated model updating method that can be used alongside a 'data lake'. The data-lake could store pooled flight test data, and if additional data is required, it could be utilised for the training of the GP model.

The disadvantage of the GP is that it is a black-box model that is not necessarily interpretable. The current approach of a flight simulator development where an initial model is created, then a flight test pilot makes an assessment on the model and the model is tuned on the suggestions of the pilot; would not be feasible with the GP model. Using the confidence levels in the predictions for the velocity and accelerations, it is still feasible that the GP would be able to achieve a higher level of fidelity compared to the physical law based model, and this has been shown in the current thesis. For future work, it may be of benefit to investigate correlations between the assessment provided by the flight test pilot and the predictive confidence associated with the GP model.

Currently, the flight test data that is collected during testing campaigns is used to refine and tune physics law based models and the flight test data is not fully exploited. The simulator

tuning process is heavily influenced by the skill and experience of those undertaking the work and by the available flight test data, therefore, the assessment process involves significant subjectivity. Utilising the GP method, would remove the subjectivity during the flight simulator development and create a more tightly-coupled link between the available flight test data and simulator development.

There are many different ways the GP could be utilised in a flight simulator, the simplest being embedding the code into the simulator and allowing the outputs from the GP predictions to drive the simulator response. Note, the figures displayed in the current thesis show 10,000 Monte Carlo runs to produce realisations. In reality, the pilot would not be able to fly 10,000 realisations simultaneously and would fly one realisation. The figures attempt to capture all the possible scenarios in which pilot could fly. Another option, is to use the GP to correct the physical law based model, as shown by the 'model error' approach presented in Chapter 9. A third method, would use the GP model or the physical law based model as a 'back up' and to step in when the primary model deviates far from the truth. For example, a GP model could be the primary and during a simulator run, if the GP model deviates far from the truth (which the confidence bounds could be used to drive this), the physical law based model could step to make the predictions. The 'back up' method differs from the 'model error' approach as during the 'model error' approach the GP would attempt to correct the physical law based method, while the 'back up' would have two interchangeable predictive models (one based on the physical law based equations and the other the GP model). The latter method could take the best properties from both GP and physical law based models.

10.3 Future Work

There are several areas where the investigation of the use of machine learning regression models for flight simulators can be pursued further. These include:

- A potential problem for the GP models that are utilising a hyperparameter for each element of the input structure is the complexity and/or size of the search space, as discussed in Chapter 6. As future work, different training sets could be utilised to change the search space, to see if this enabled to GPs models to create comparable predictions to

the GP utilising one hyperparameter. Another option, is to run the optimisation process many times, but this may not solve the issue, as the number of runs needed to 'explore' the space might be too large to be practical.

- The sparse GP results presented in the current thesis used training points that were part of the original dataset; this may have had a negative impact on the performance of the GP model. In future work, one could investigate utilising training points that are not part of the original dataset.

- The GPs in the current thesis are trained and tested on a specific manoeuvre (3-2-1-1 for the pitch rate, roll rate, yaw rate and heave). It would be beneficial to train the model on a full mission or data from a flight test campaign to investigate if the GP could accurately predict a wide range of manoeuvres, utilising the confidence bounds to assess if more data is required in specific regions.

- As discussed in Chapter 6, the GP models displayed generalise poorly to other manoeuvres. To increase the chance that the GP would be able to predict different manoeuvres, more training data would be required. As previously mentioned, this would increase the time complexity of the training phase. It would be beneficial to employ the sparse GP method on larger datasets in the future and potentially on a full flight test campaign.

- The GPs predicting the model error in Chapter 9 used an already tuned FLIGHTLAB model. In future work, it could be beneficial to tune the FLIGHTLAB model at the same time as optimising the GP model in an attempt to increase the accuracy of the grey-box model, as described in the study by O'Hagan [72, 74]

- One feature of rotorcraft dynamics is that the output responses are coupled; therefore, it may be beneficial to create a multi-input-multi-output (MIMO) GP model. The MIMO GP may have the potential to increase the accuracy of the predictions due to the inclusion of the cross-coupling aspect. The missing coupling dynamics could be a reason why the realisations for the FMP do not encompass all of the real flight test data of the Bo105 rotorcraft. There has been work on GP with multiple outputs [77, 78, 79, 80, 81], and

work on multi-output sparse GP regression [82]. The study by Zhao and Sun [83] uses a multi-output GP on dynamical systems including motion capture, traffic flow and robot dynamics.

- To investigate the best way of deploying the GP model into the flight simulator. These options include, using the GP to drive the responses, the grey-box model option (where the GP corrects the physical law based model), and the final option would be to use one model (GP or physical law based model) as the primary and the other as a backup for when the primary deviates far from the truth, the backup could step in.

- An investigation of the correlations between the assessment provided by the flight test pilot and the predictive confidence associated with the GP model.

Appendix A

Feasibility Study Bo105 data

A.1 Yaw Rate

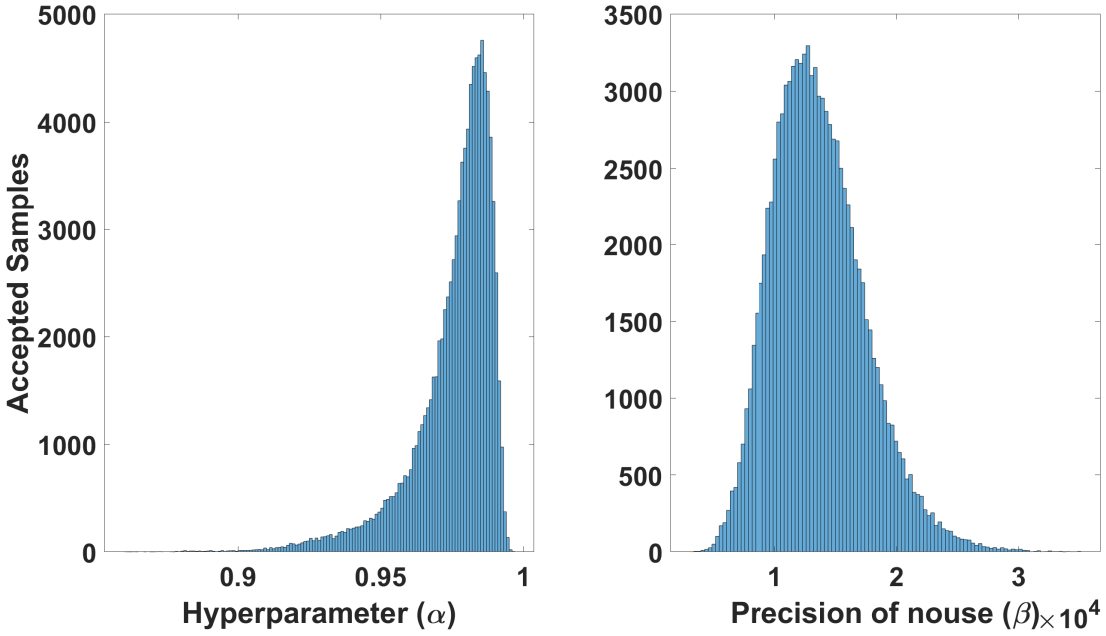


Figure A.1: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing the current longitudinal pilot input and one lagged yaw rate term.

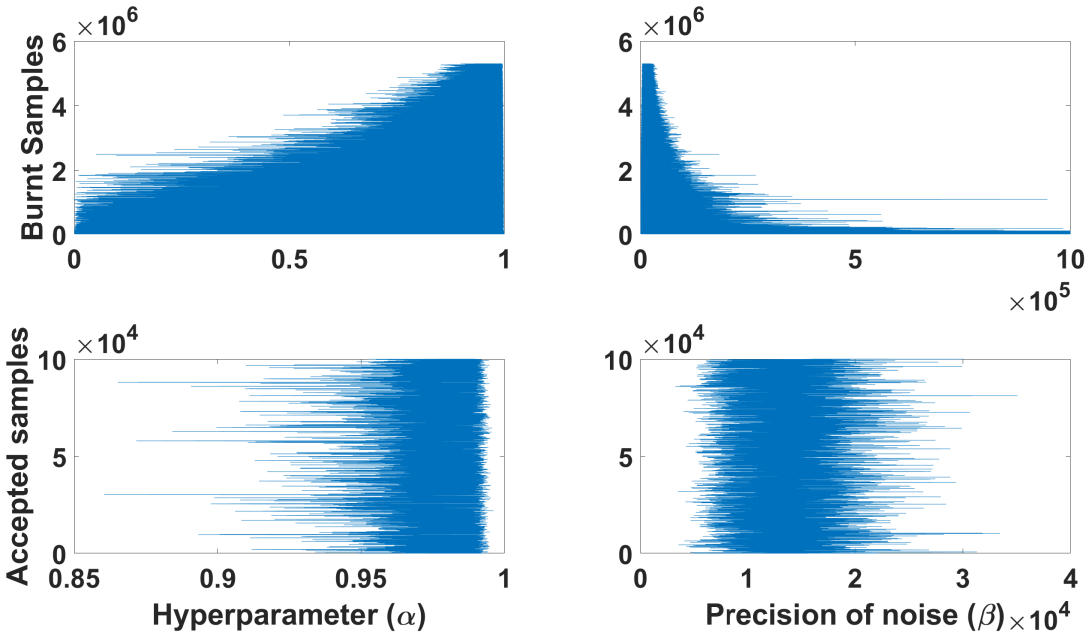


Figure A.2: Simulated annealing burn-in and accepted samples used to create the yaw rate Gaussian process model using the input structure containing the current longitudinal pilot input and one lagged yaw rate term.

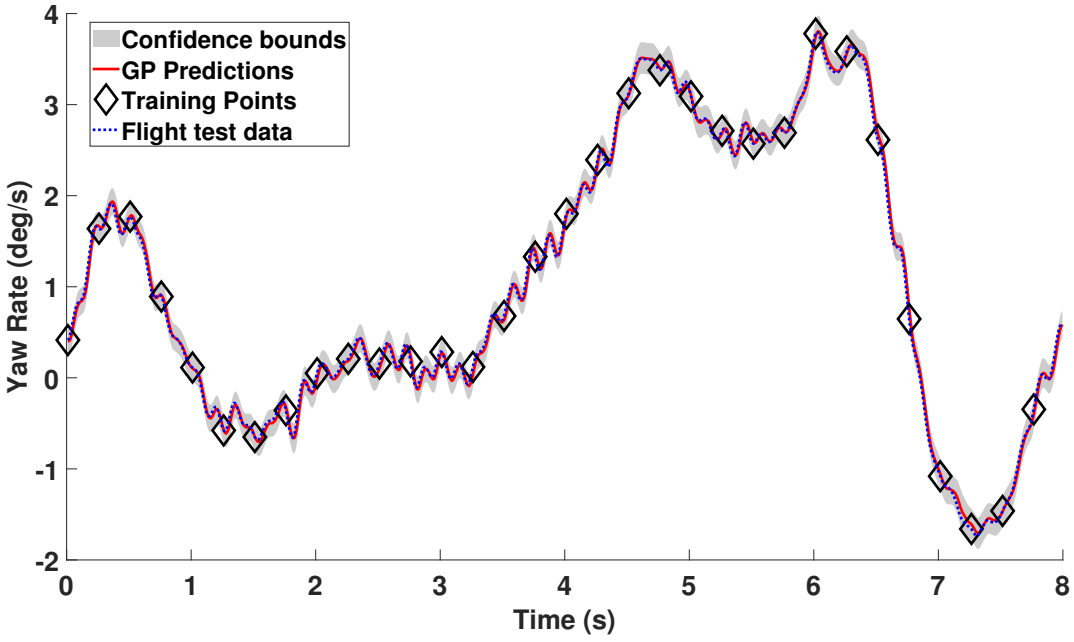


Figure A.3: Gaussian process one step ahead yaw rate predictions using the input structure containing the current longitudinal pilot input and one lagged yaw rate term using the hyper-parameters located by the simulated annealing results shown in Figure A.1.

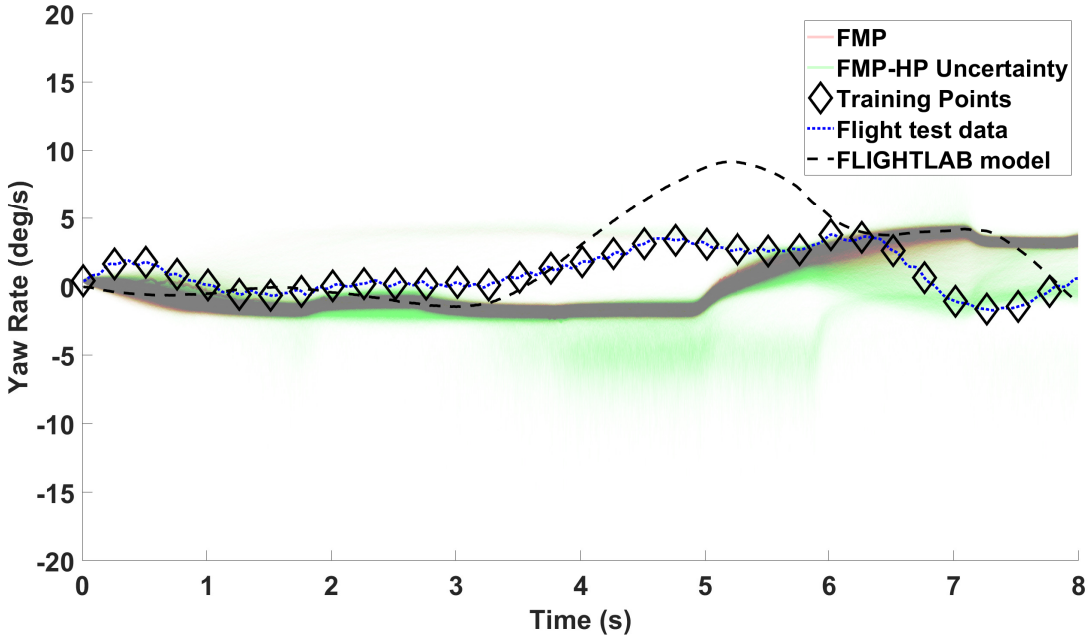


Figure A.4: Realisations of the Gaussian process yaw rate full model prediction using the input structure containing the current longitudinal pilot input and one lagged yaw rate term using the hyperparameters located by the simulated annealing results shown in Figure A.1, with a comparison to the corresponding FLIGHTLAB model.

A.2 Heave

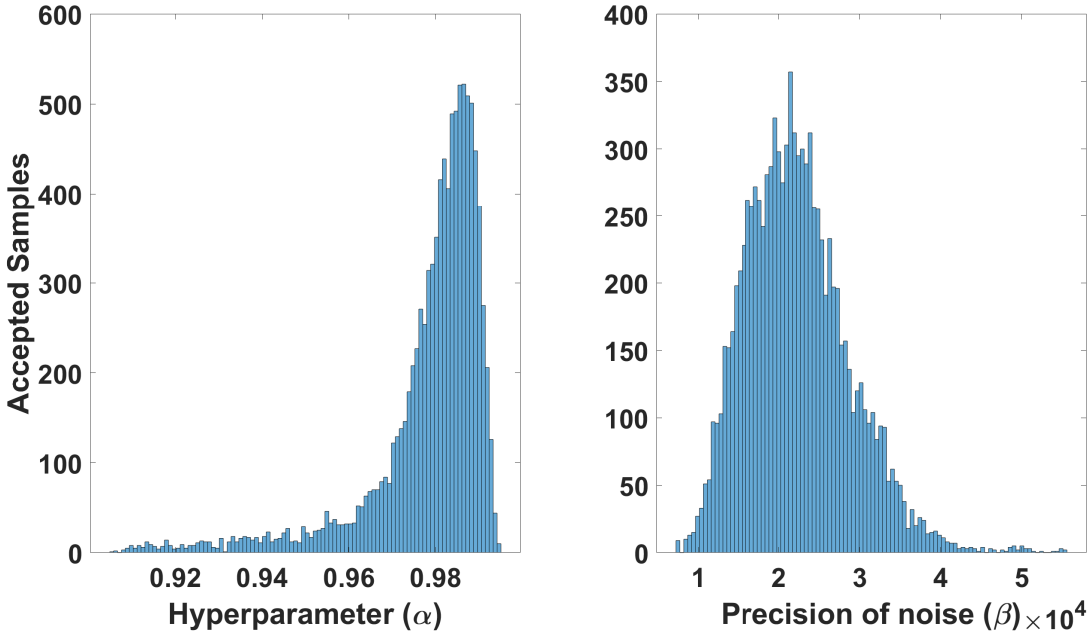


Figure A.5: Simulated annealing results used to create the heave Gaussian process model using the input structure containing the current longitudinal pilot input and one lagged heave term.

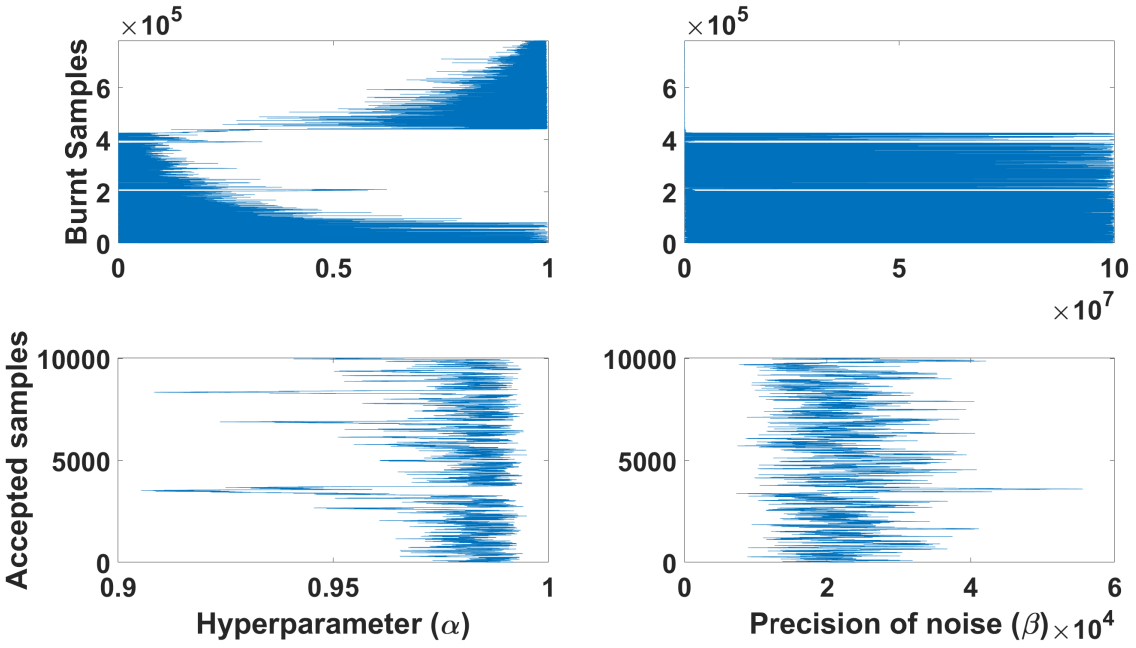


Figure A.6: Simulated annealing burn-in and accepted samples used to create the heave Gaussian process model using the input structure containing the current longitudinal pilot input and one lagged heave term.

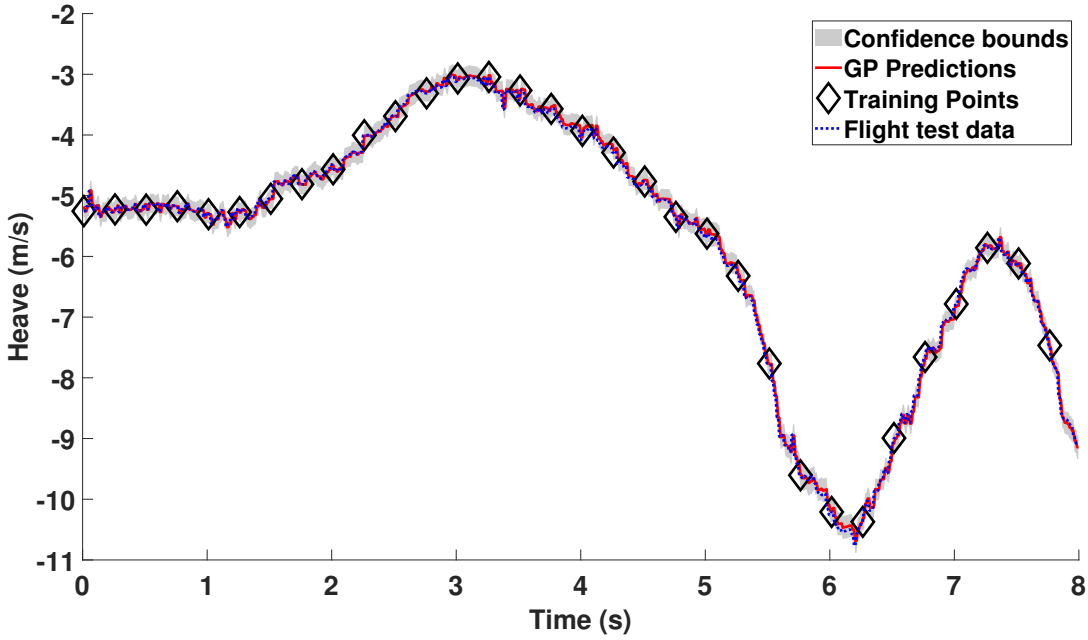


Figure A.7: Gaussian process one step ahead heave predictions using the input structure containing the current longitudinal pilot input and one lagged heave term using the hyperparameters located by the simulated annealing results shown in Figure A.5.

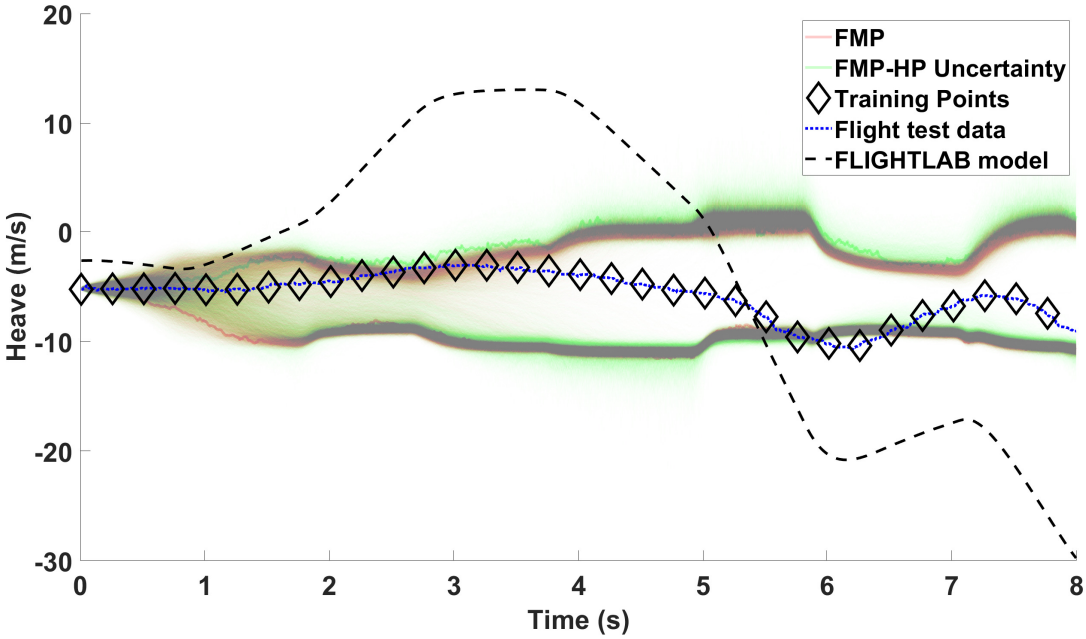


Figure A.8: Realisations of the Gaussian process heave full model prediction using the input structure containing the current longitudinal pilot input and one lagged heave term using the hyperparameters located by the simulated annealing results shown in Figure A.5, with a comparison to the corresponding FLIGHTLAB model.

Appendix B

Investigation of GP input structures

B.1 Pitch Rate

Figures in this Section relate to the pitch rate response GP results shown in Section 5.1.1.

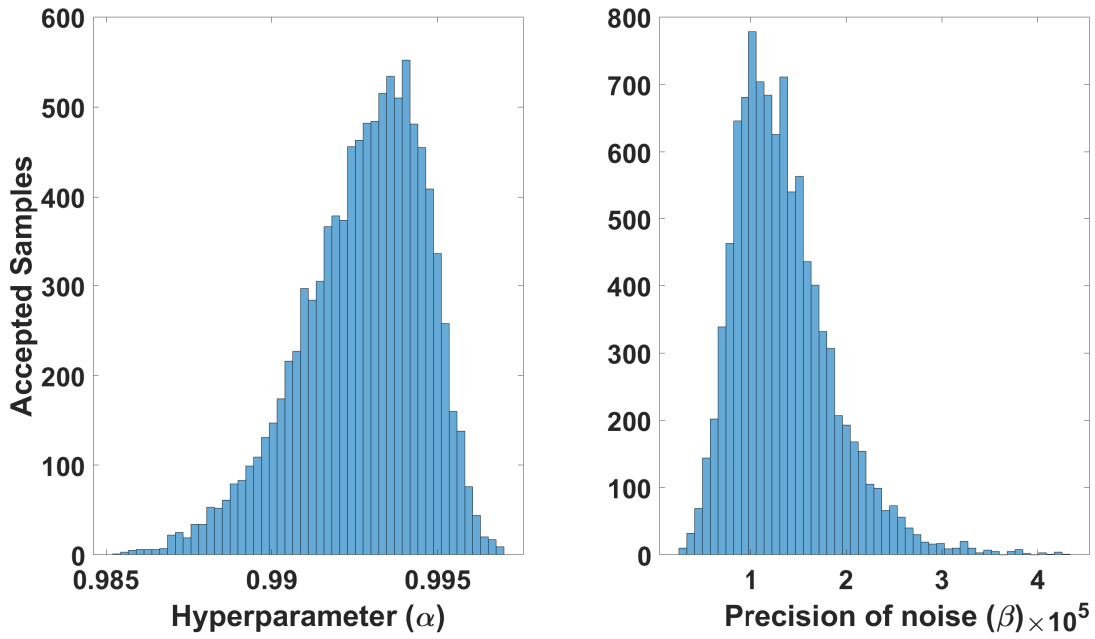


Figure B.1: Simulated annealing results used to create the pitch rate Gaussian process model using the input structure containing all of the current pilot inputs and one lagged pitch rate term.

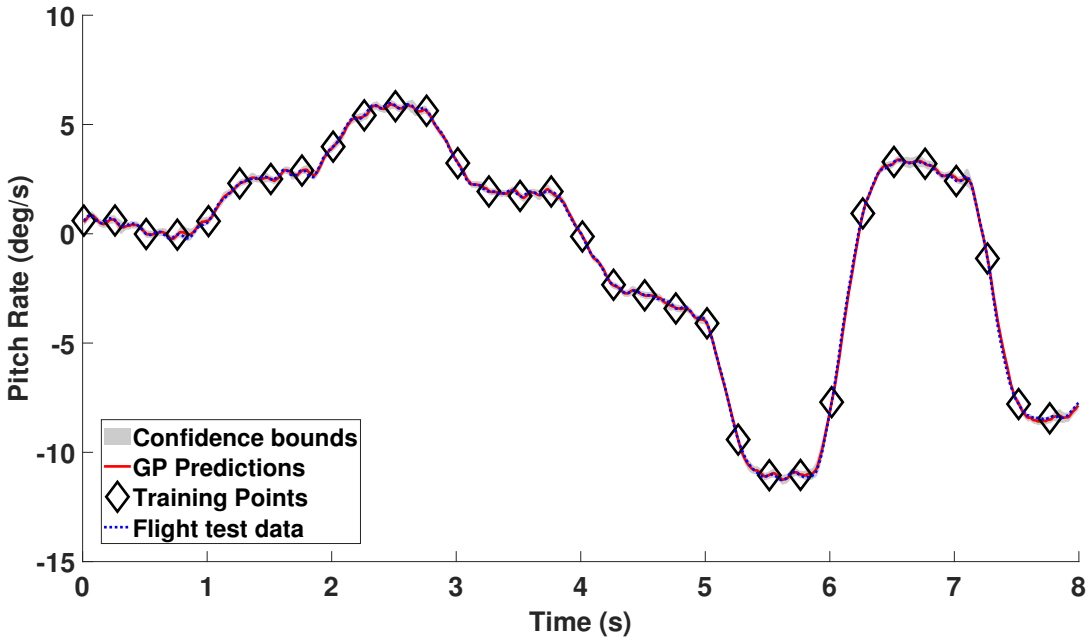


Figure B.2: Gaussian process one step ahead pitch rate predictions using the input structure containing all of the current pilot inputs and one lagged pitch rate term using the hyperparameters located by the simulated annealing results shown in Figure B.1.

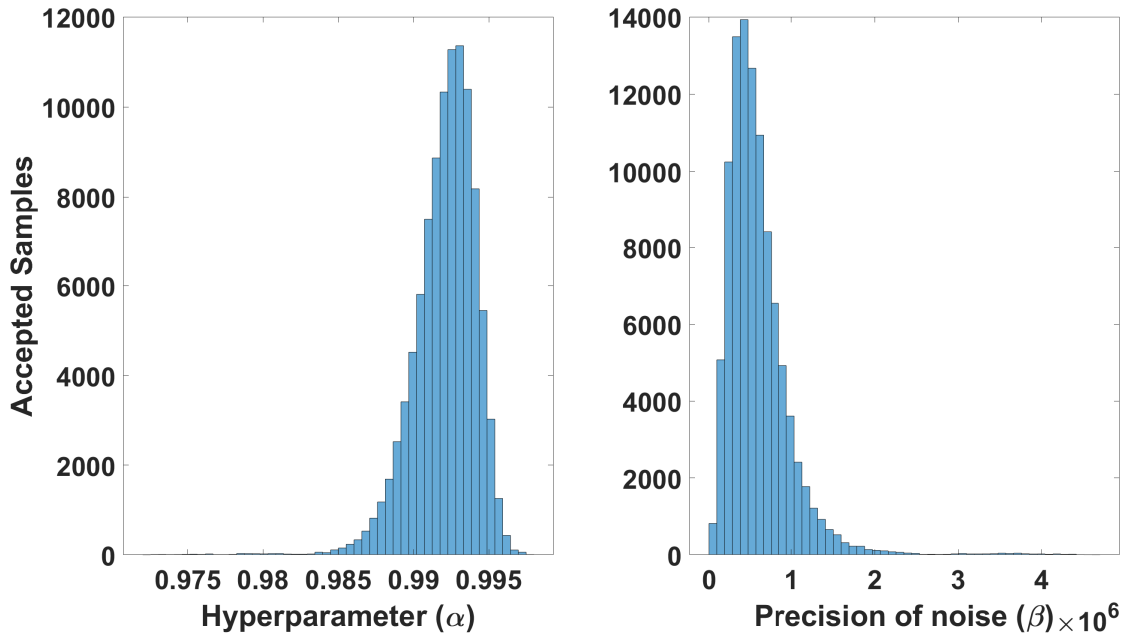


Figure B.3: Simulated annealing results used to create the pitch rate Gaussian process model using the input structure containing all of the current pilot inputs and two lagged pitch rate terms.

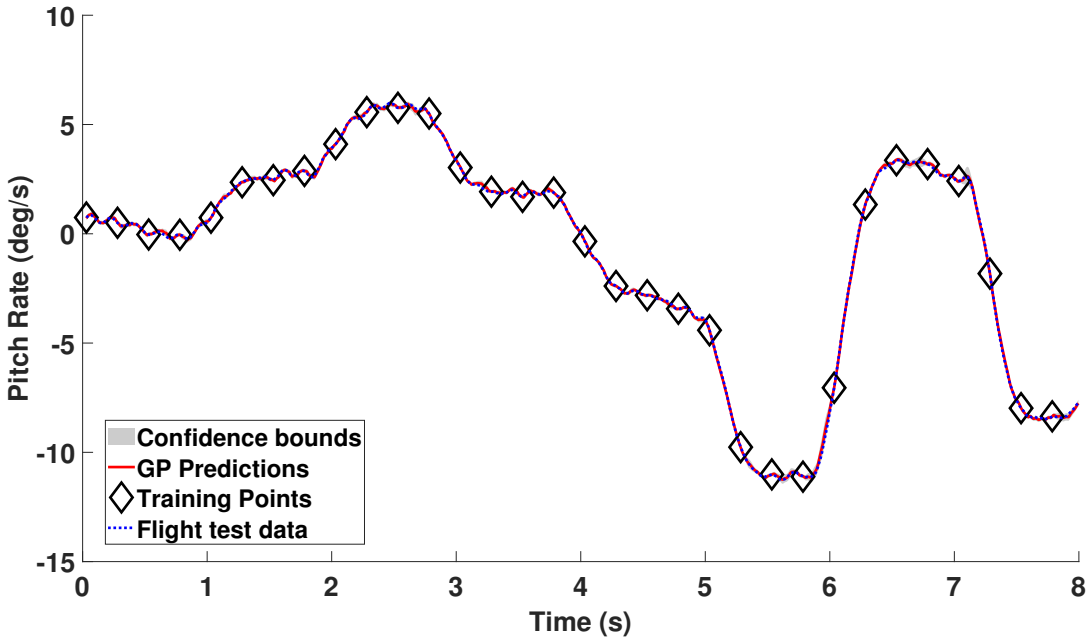


Figure B.4: Gaussian process one step ahead pitch rate predictions using the input structure containing all of the current pilot inputs and two lagged pitch rate term using the hyperparameters located by the simulated annealing results shown in Figure B.3.

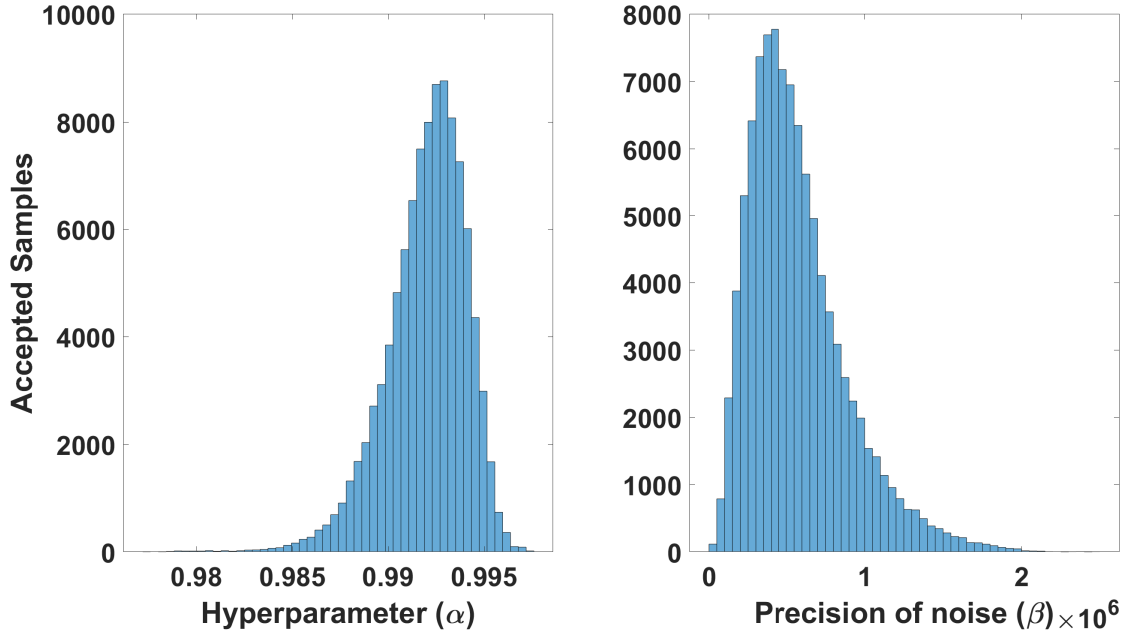


Figure B.5: Simulated annealing results used to create the pitch rate Gaussian process model using the input structure containing all of the current pilot inputs and three lagged pitch rate terms.

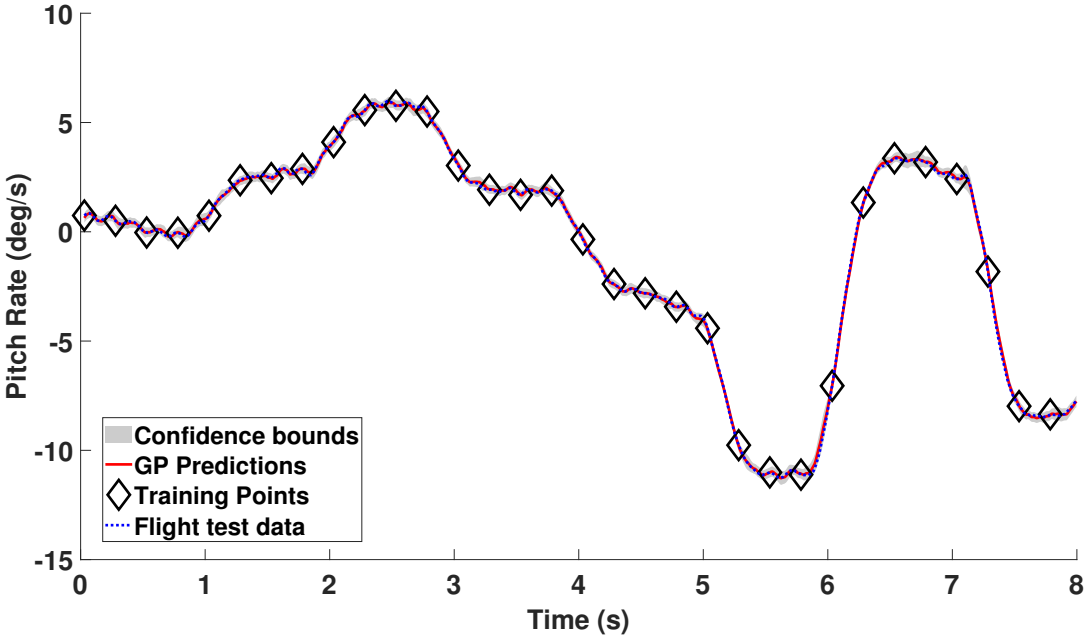


Figure B.6: Gaussian process one step ahead pitch rate predictions using the input structure containing all of the current pilot inputs and three lagged pitch rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.5.

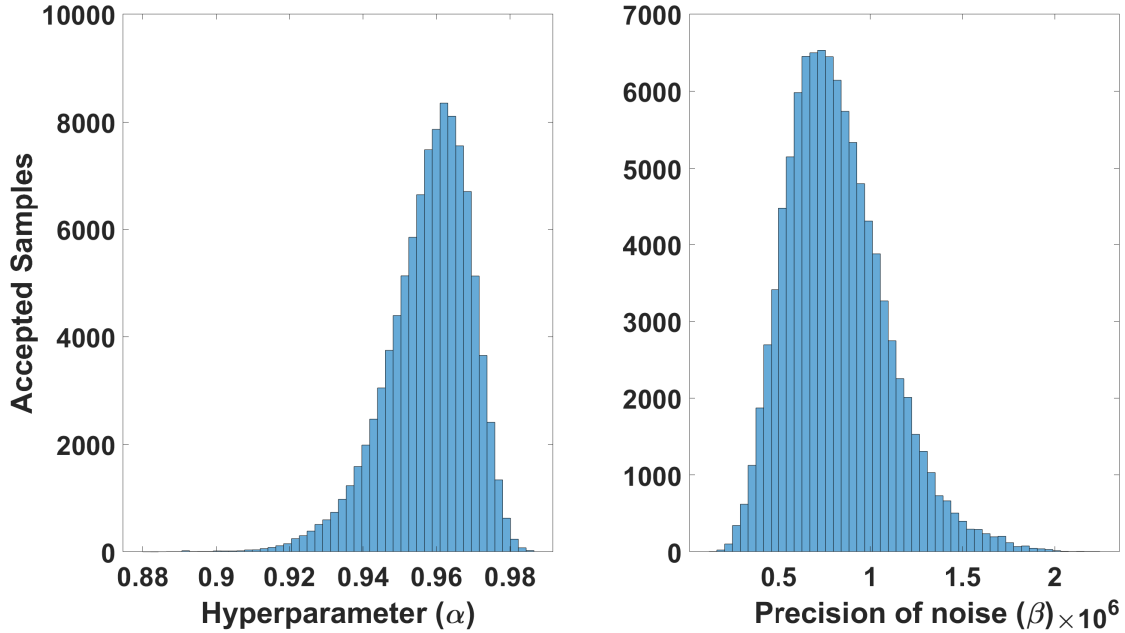


Figure B.7: Simulated annealing results used to create the pitch rate Gaussian process model using the input structure containing the current longitudinal pilot input and two lagged pitch rate terms.

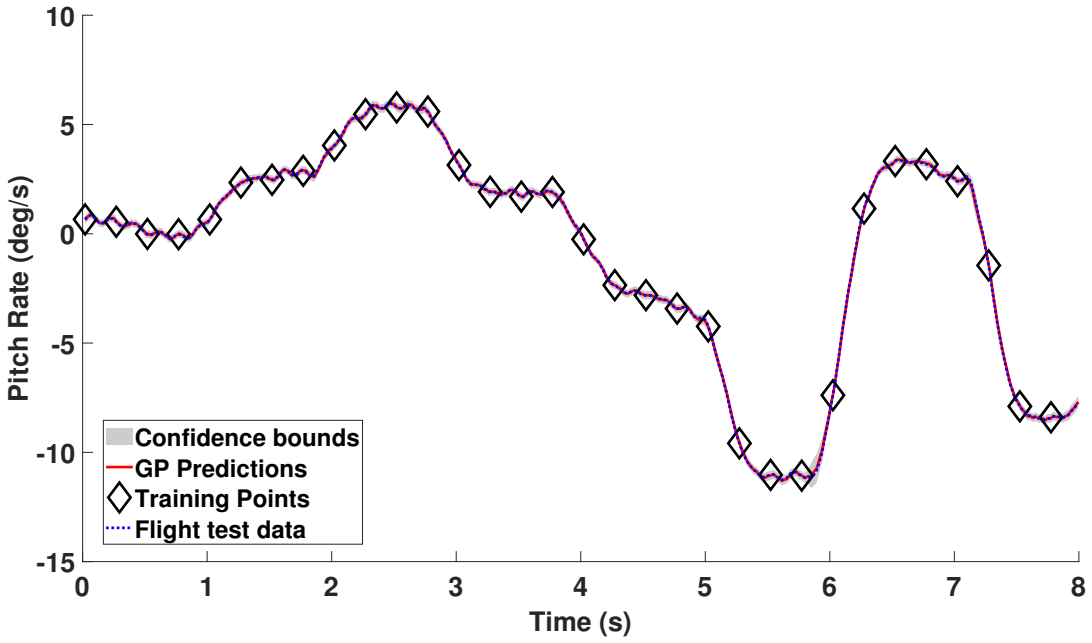


Figure B.8: Gaussian process one step ahead pitch rate predictions using the input structure containing the current longitudinal pilot input and two lagged pitch rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.7.

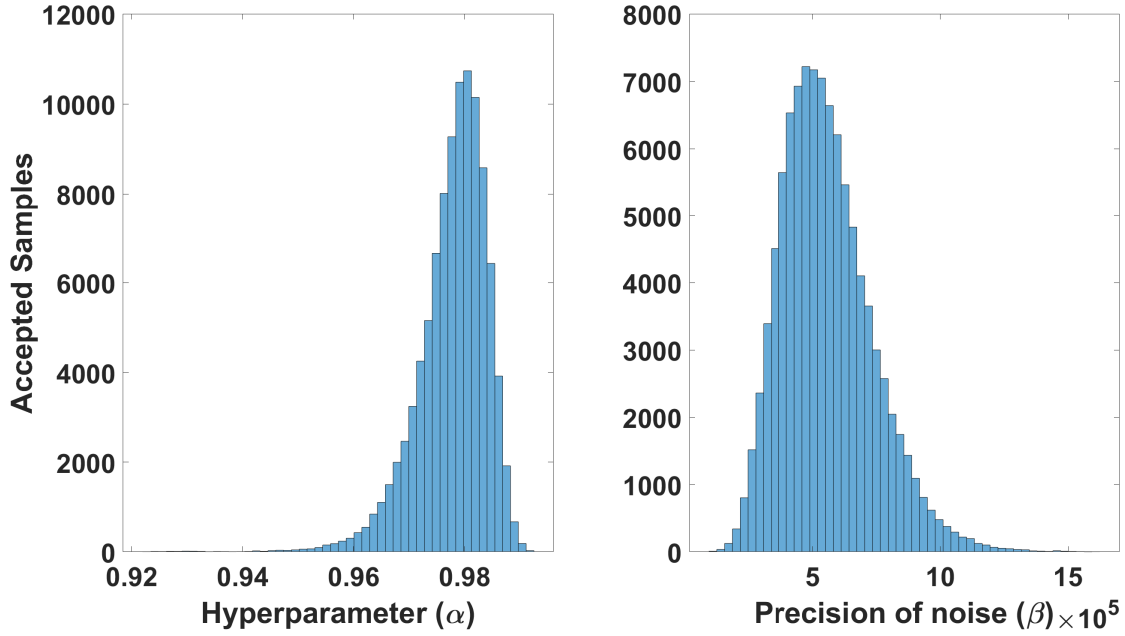


Figure B.9: Simulated annealing results used to create the pitch rate Gaussian process model using the input structure containing the current longitudinal pilot input and three lagged pitch rate terms.

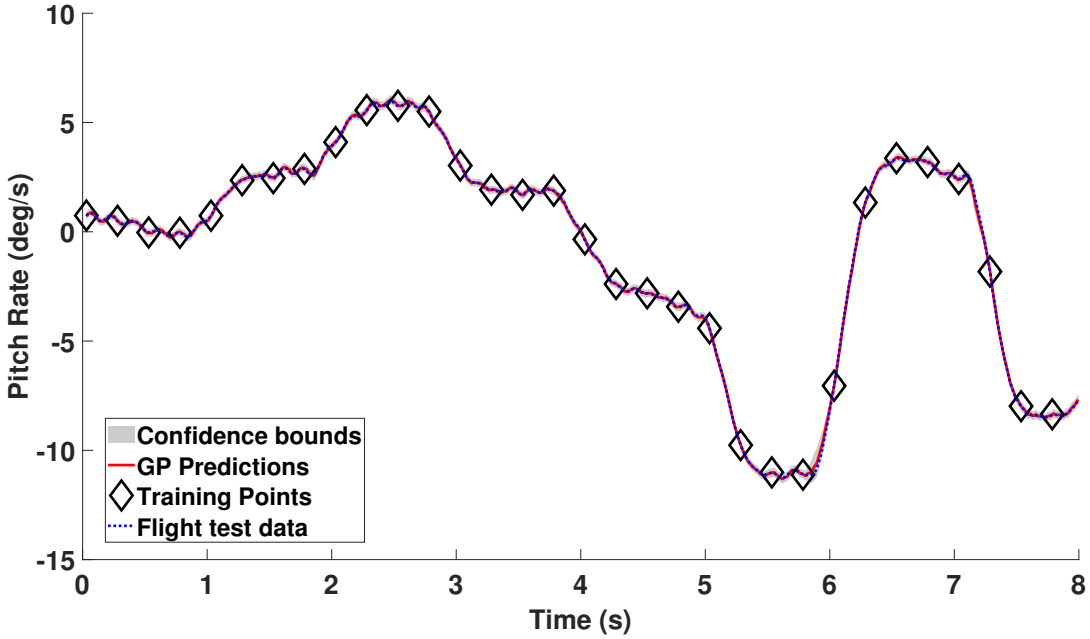


Figure B.10: Gaussian process one step ahead pitch rate predictions using the input structure containing the current longitudinal pilot input and three lagged pitch rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.9.

B.2 Roll Rate

Figures in this Section relate to the roll rate response GP results shown in Section 5.1.2.

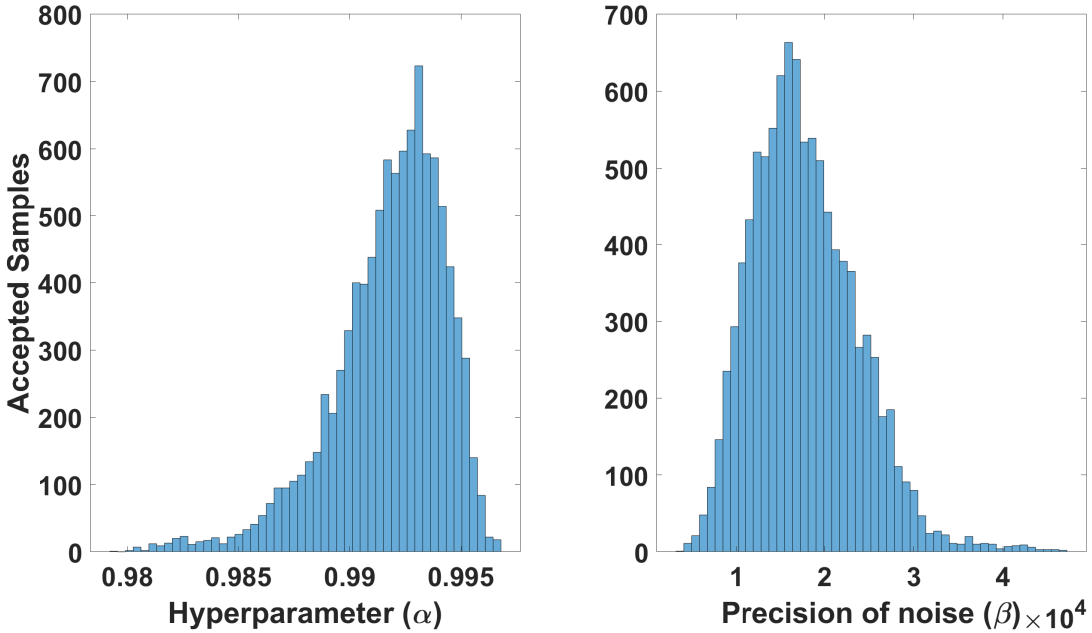


Figure B.11: Simulated annealing results used to create the roll rate Gaussian process model using the input structure containing all of the current pilot inputs and one lagged roll rate term.

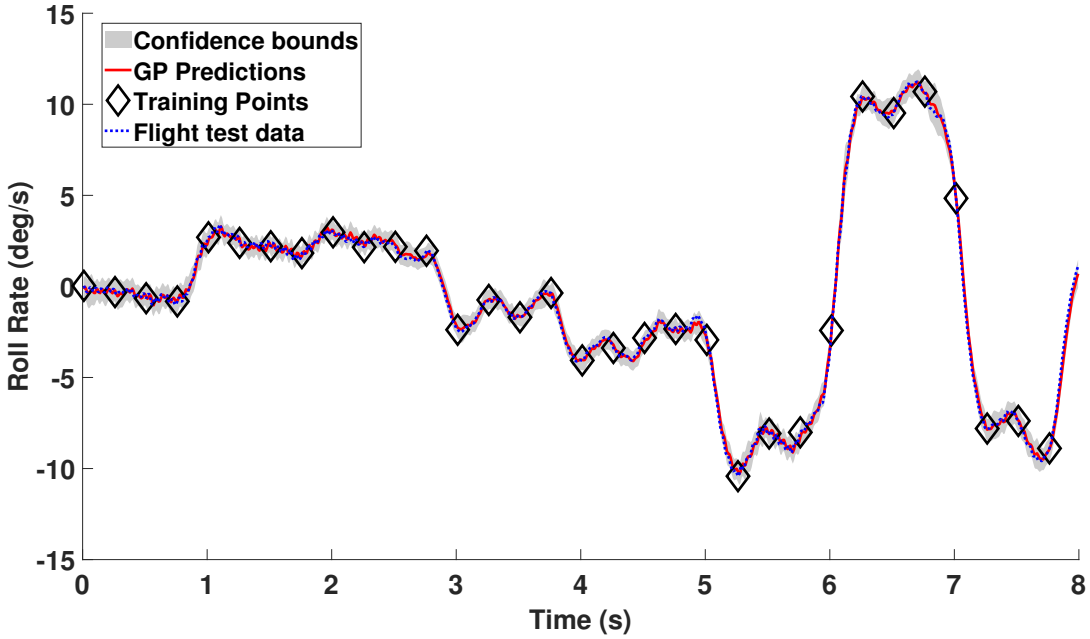


Figure B.12: Gaussian process one step ahead roll rate predictions using the input structure containing all of the current pilot inputs and one lagged roll rate term using the hyperparameters located by the simulated annealing results shown in Figure B.11.

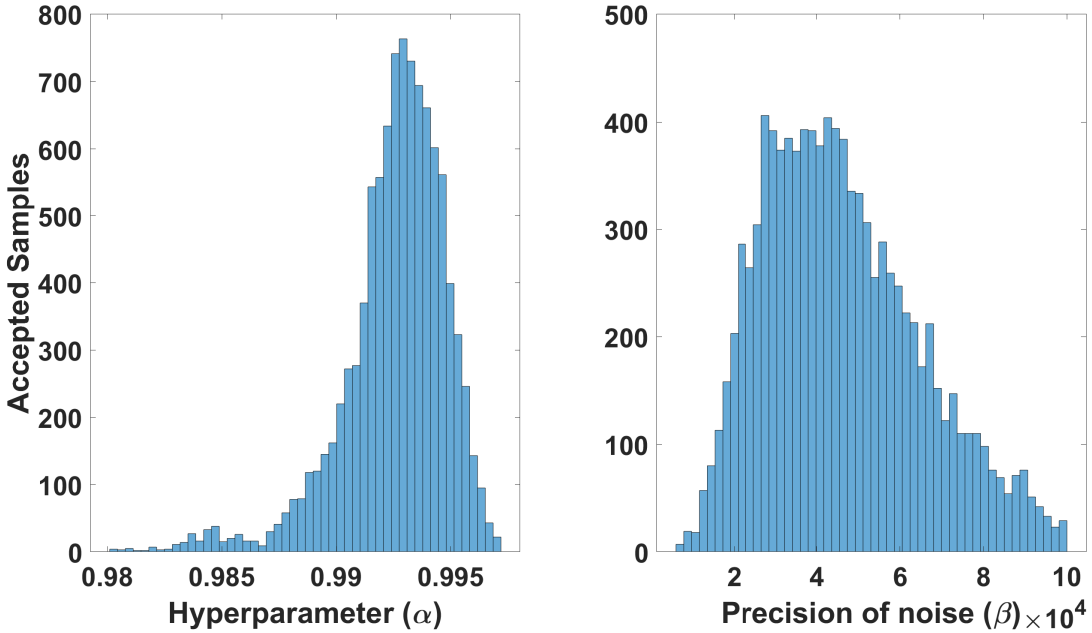


Figure B.13: Simulated annealing results used to create the roll rate Gaussian process model using the input structure containing all of the current pilot inputs and two lagged roll rate terms.

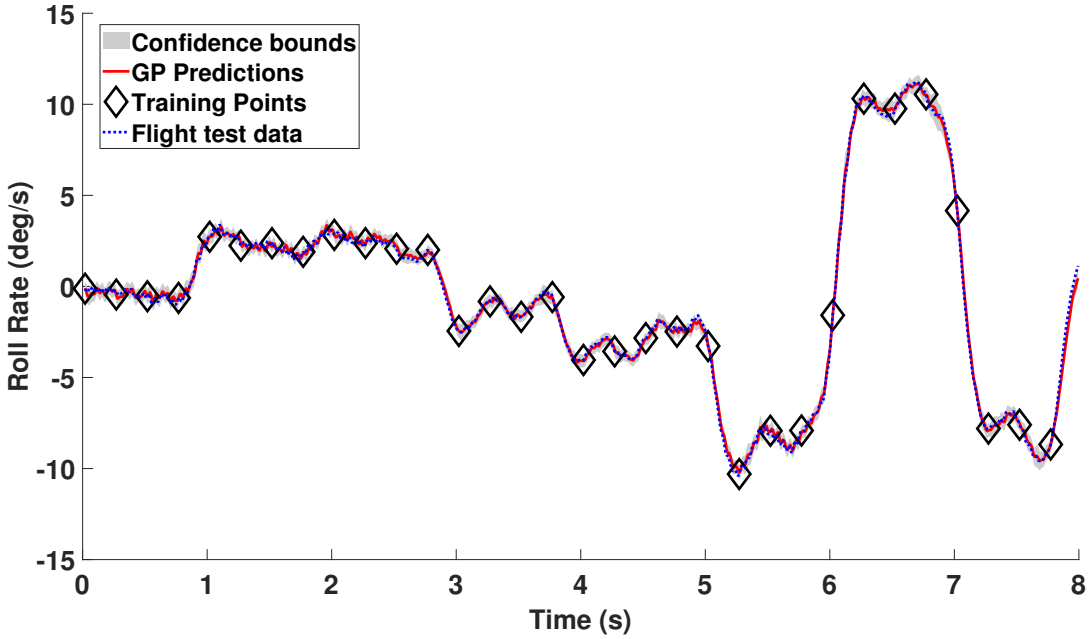


Figure B.14: Gaussian process one step ahead roll rate predictions using the input structure containing all of the current pilot inputs and two lagged roll rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.13.

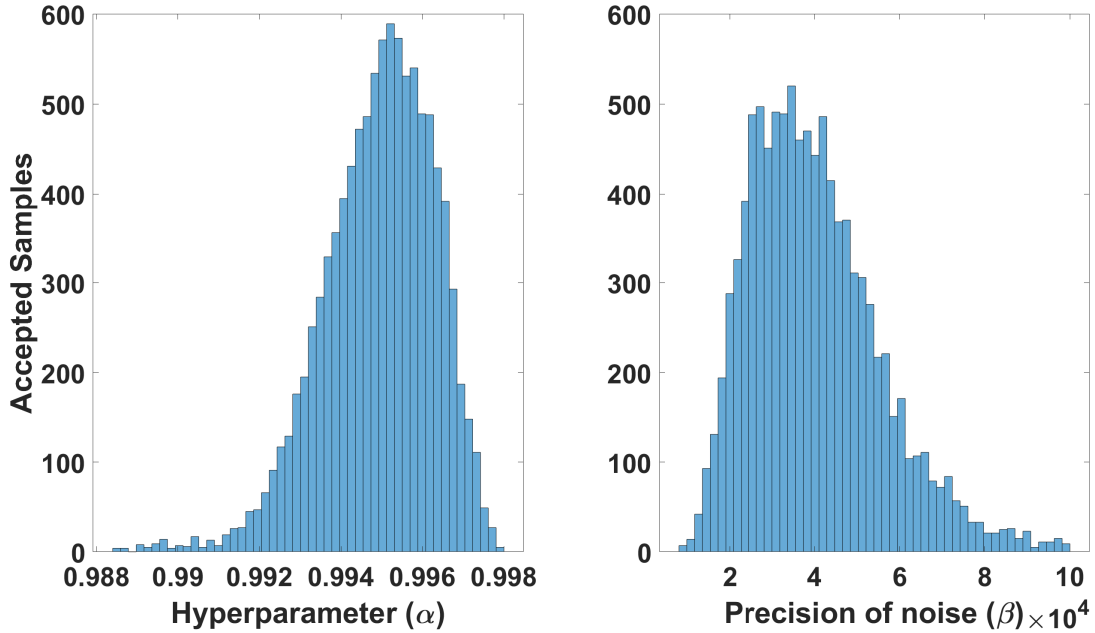


Figure B.15: Simulated annealing results used to create the roll rate Gaussian process model using the input structure containing all of the current pilot inputs and three lagged roll rate terms.

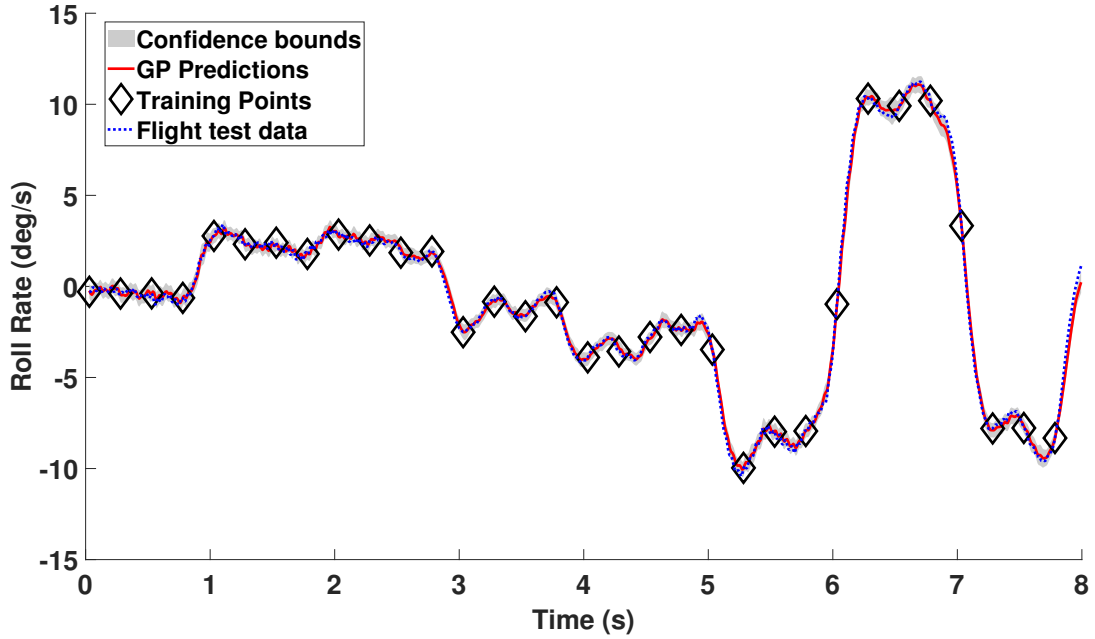


Figure B.16: Gaussian process one step ahead roll rate predictions using the input structure containing all of the current pilot inputs and three lagged roll rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.15.

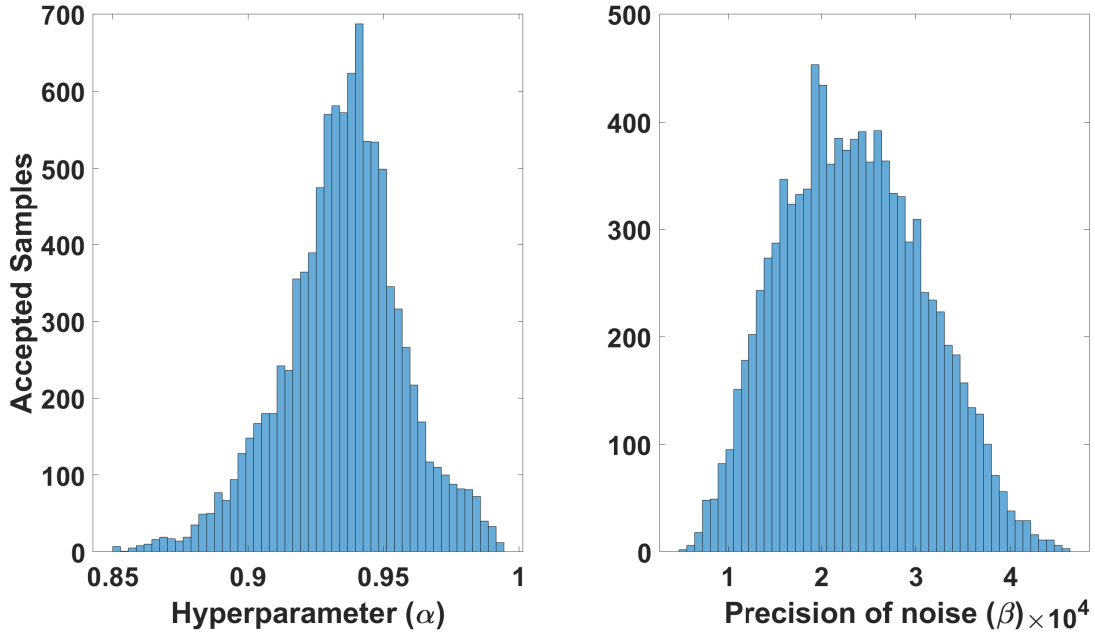


Figure B.17: Simulated annealing results used to create the roll rate Gaussian process model using the input structure containing the current lateral pilot input and one lagged roll rate term.

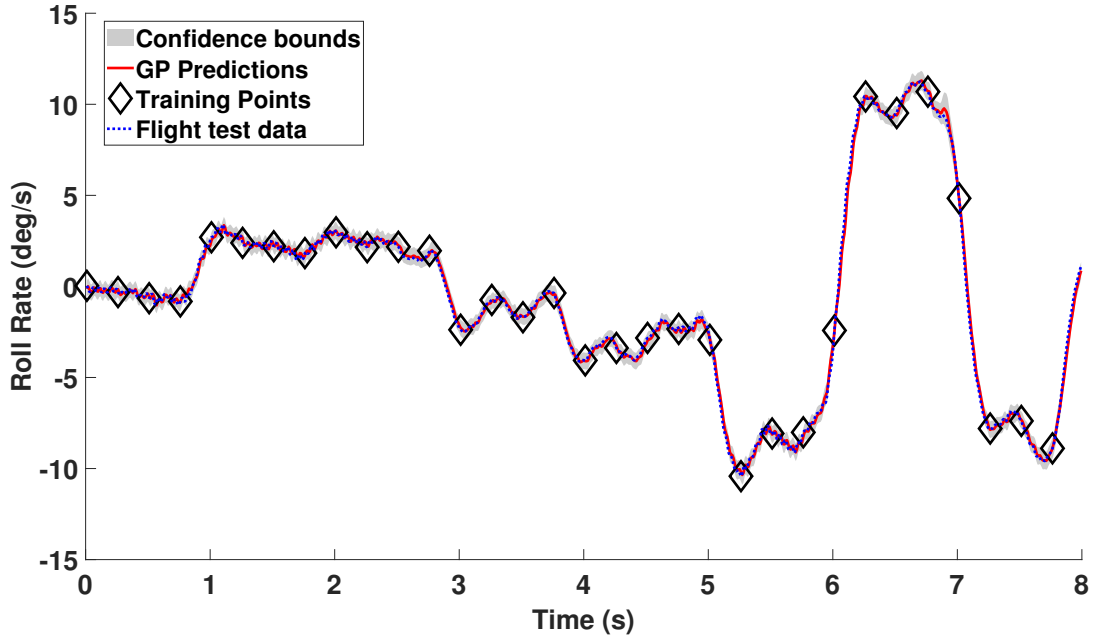


Figure B.18: Gaussian process one step ahead roll rate predictions using the input structure containing the current lateral pilot input and one lagged roll rate term using the hyperparameters located by the simulated annealing results shown in Figure B.17.

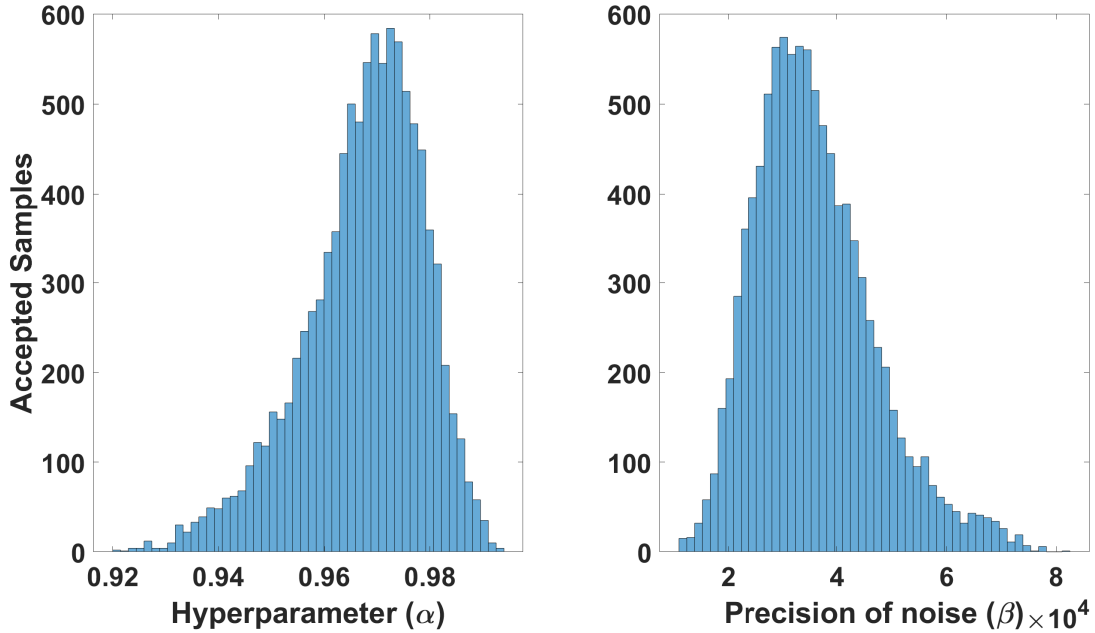


Figure B.19: Simulated annealing results used to create the roll rate Gaussian process model using the input structure containing the current lateral pilot input and two lagged roll rate terms.

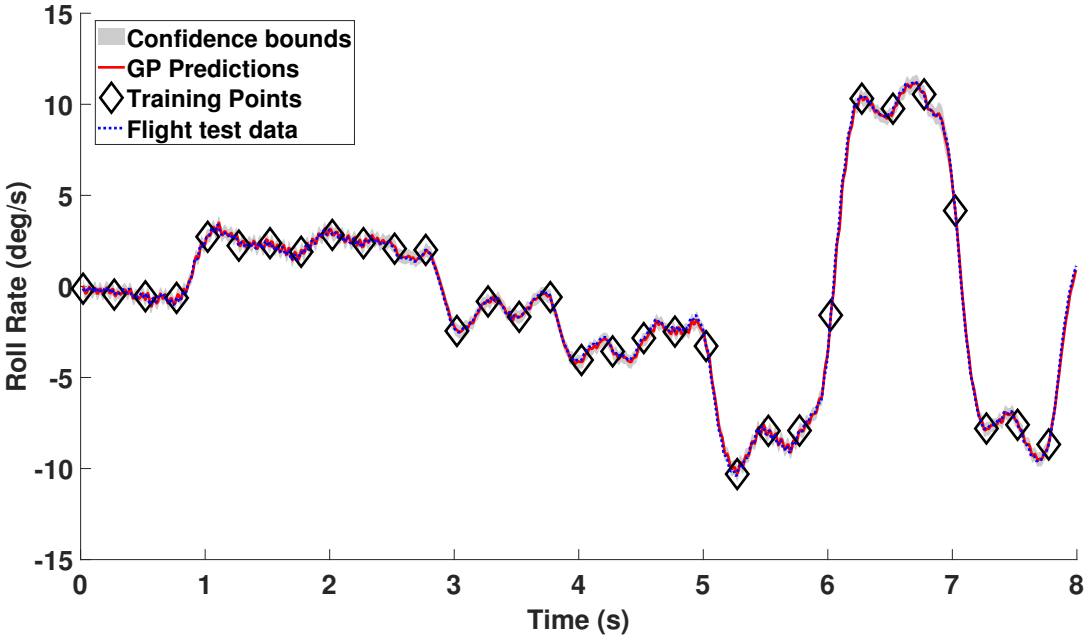


Figure B.20: Gaussian process one step ahead roll rate predictions using the input structure containing the current lateral pilot input and two lagged roll rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.19.

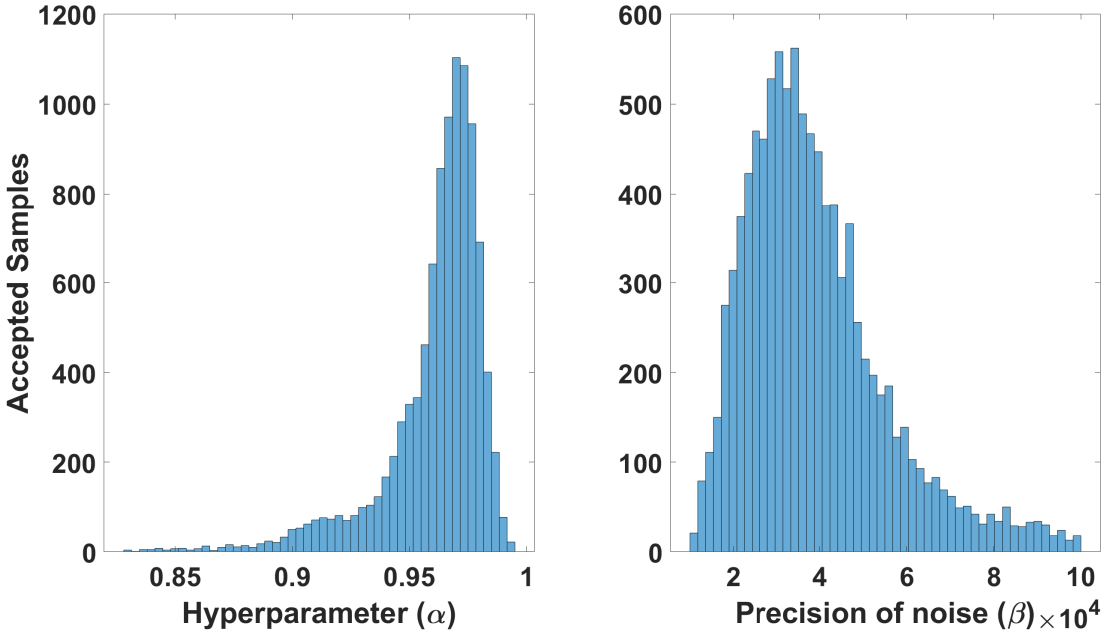


Figure B.21: Simulated annealing results used to create the roll rate Gaussian process model using the input structure containing the current lateral pilot input and three lagged roll rate terms.

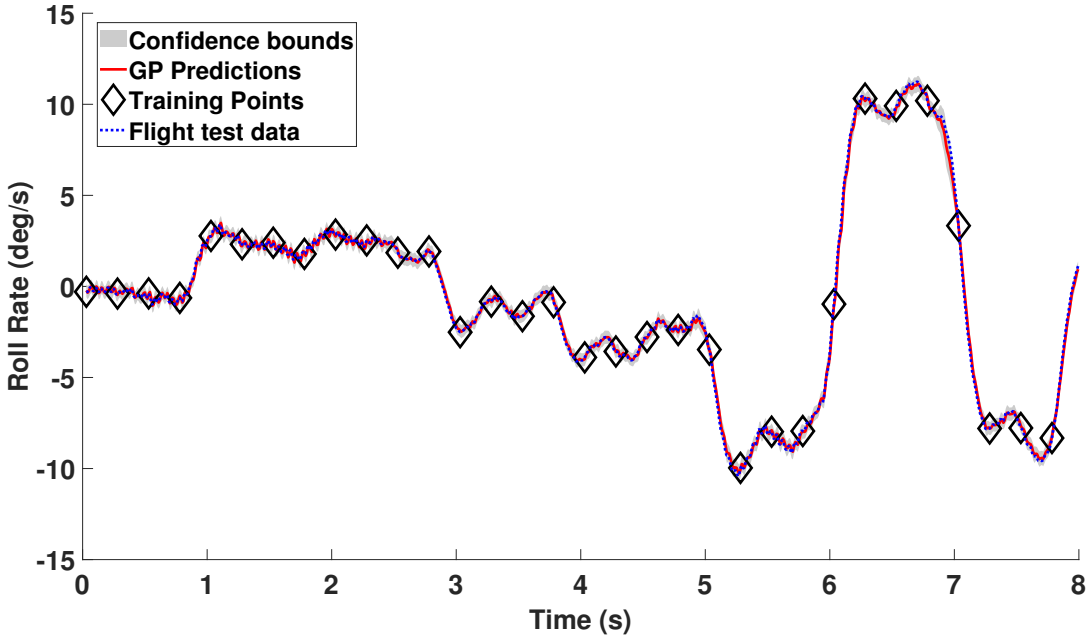


Figure B.22: Gaussian process one step ahead roll rate predictions using the input structure containing the current lateral pilot input and three lagged roll rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.21.

B.3 Yaw Rate

Figures in this Section relate to the yaw rate response GP results shown in Section 5.1.3.

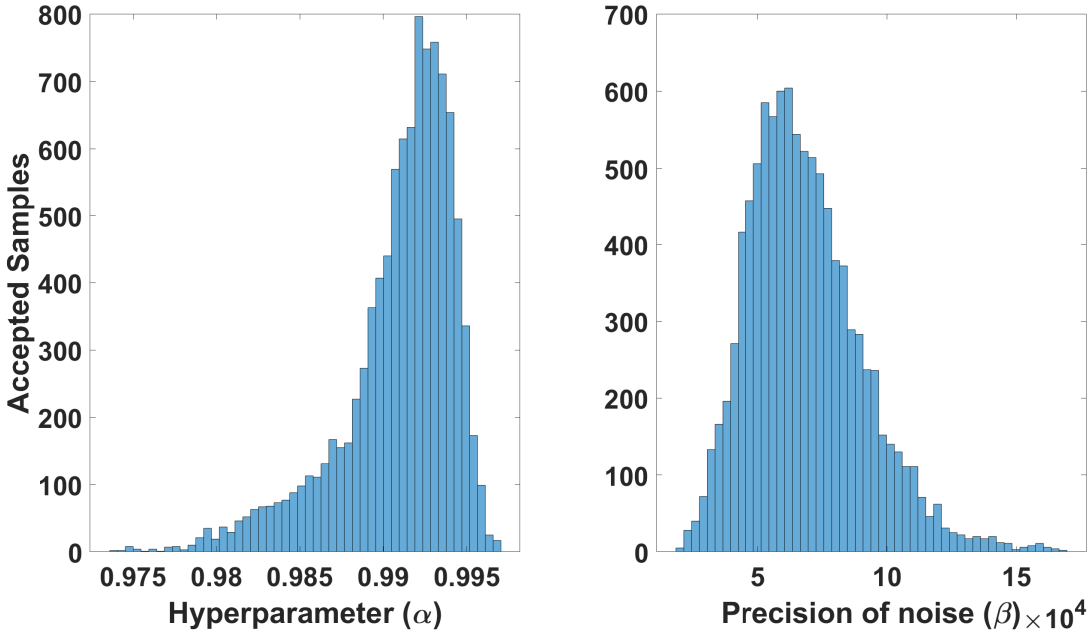


Figure B.23: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing all of the current pilot inputs and one lagged yaw rate term.

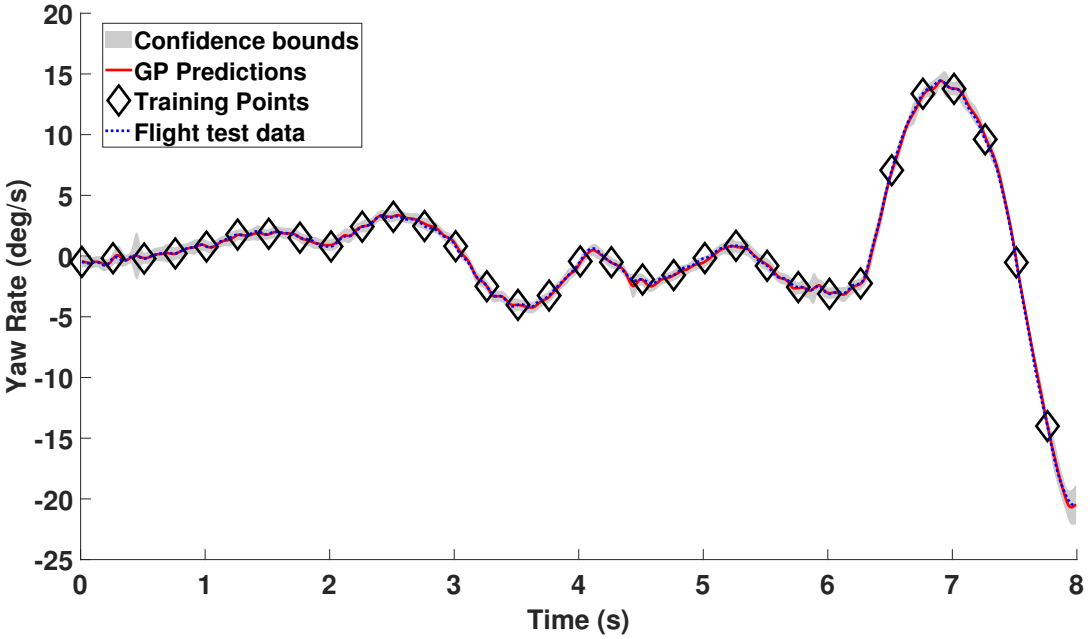


Figure B.24: Gaussian process one step ahead yaw rate predictions using the input structure containing all of the current pilot inputs and one lagged yaw rate term using the hyperparameters located by the simulated annealing results shown in Figure B.23.

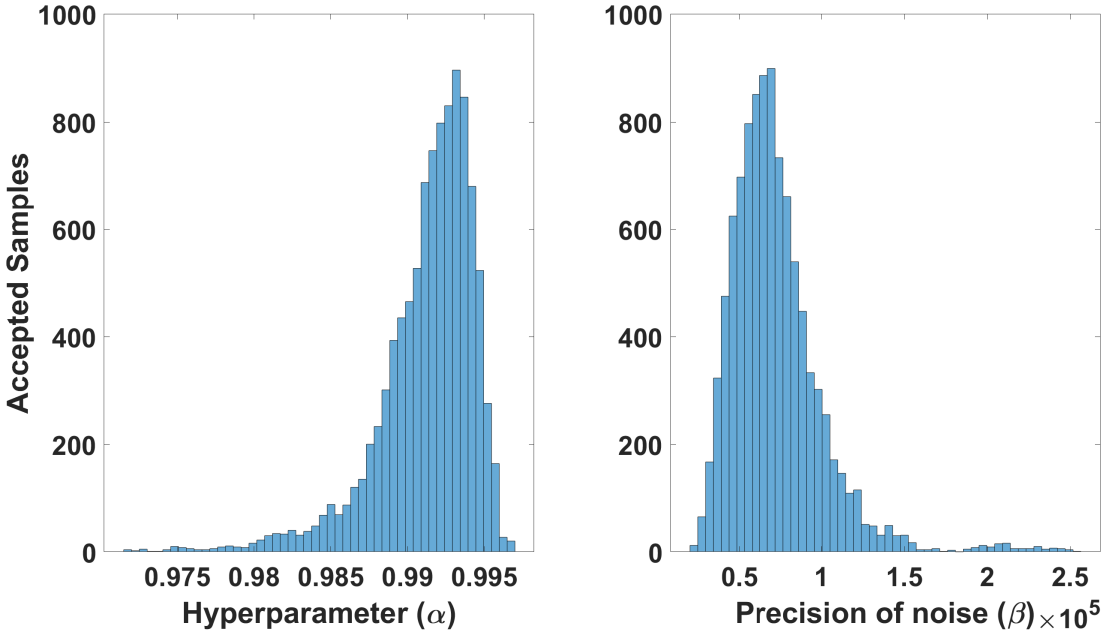


Figure B.25: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing all of the current pilot inputs and two lagged yaw rate terms.

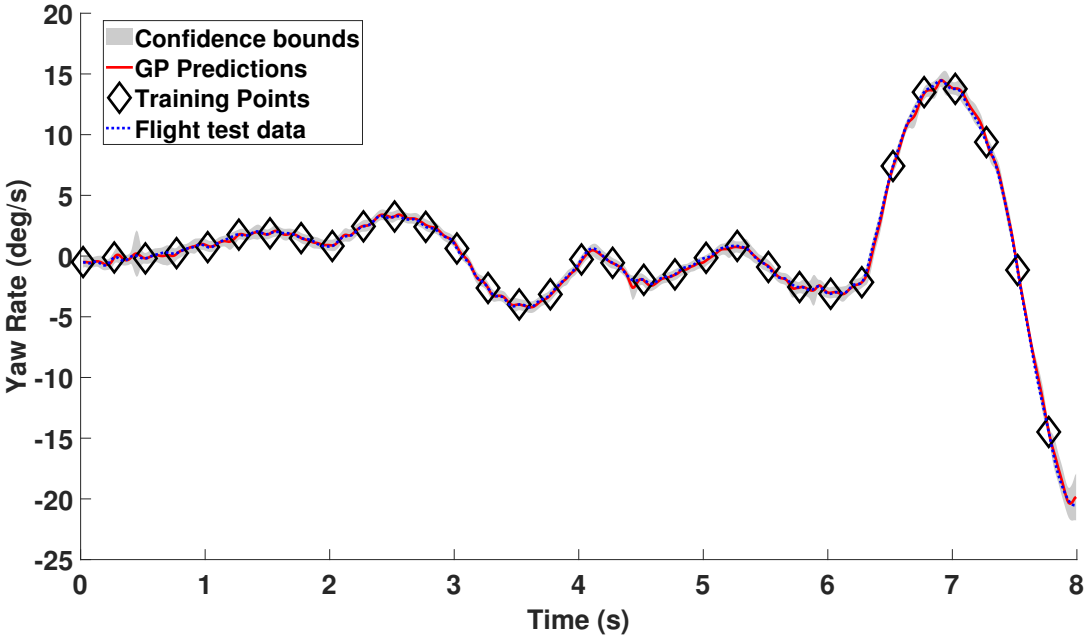


Figure B.26: Gaussian process one step ahead yaw rate predictions using the input structure containing all of the current pilot inputs and two lagged yaw rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.25.

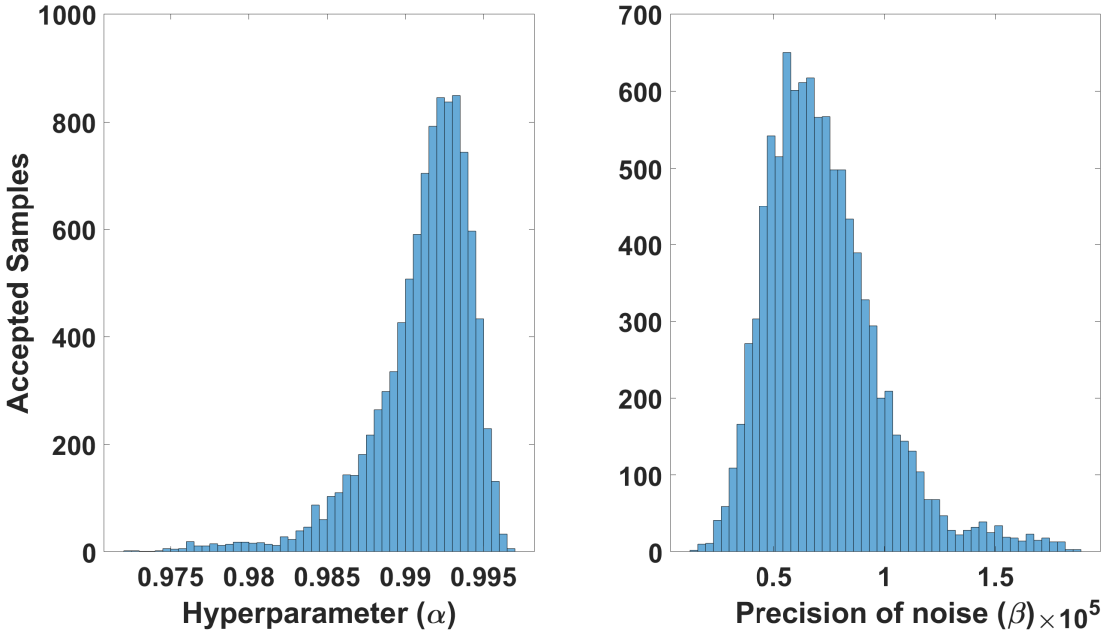


Figure B.27: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing all of the current pilot inputs and three lagged yaw rate terms.

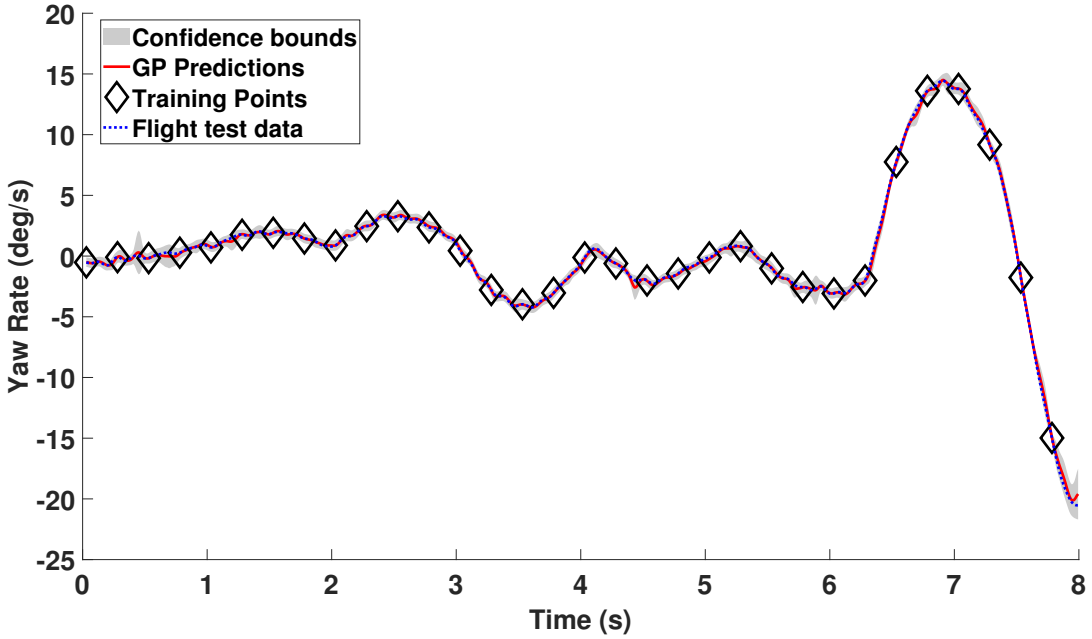


Figure B.28: Gaussian process one step ahead yaw rate predictions using the input structure containing all of the current pilot inputs and three lagged yaw rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.27.

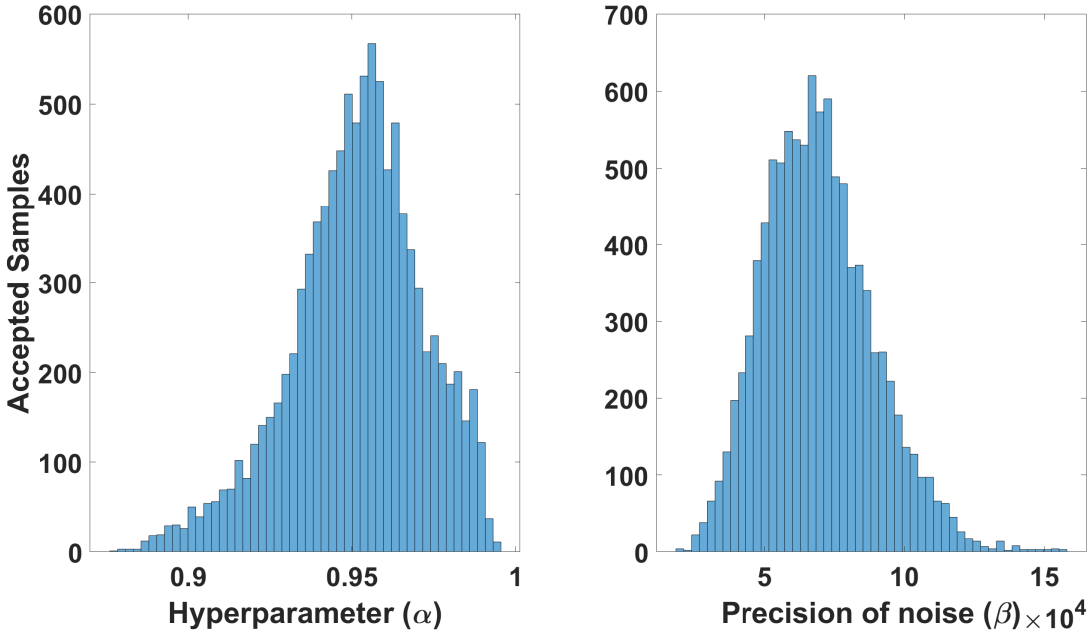


Figure B.29: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing the current pedal pilot input and one lagged yaw rate term.

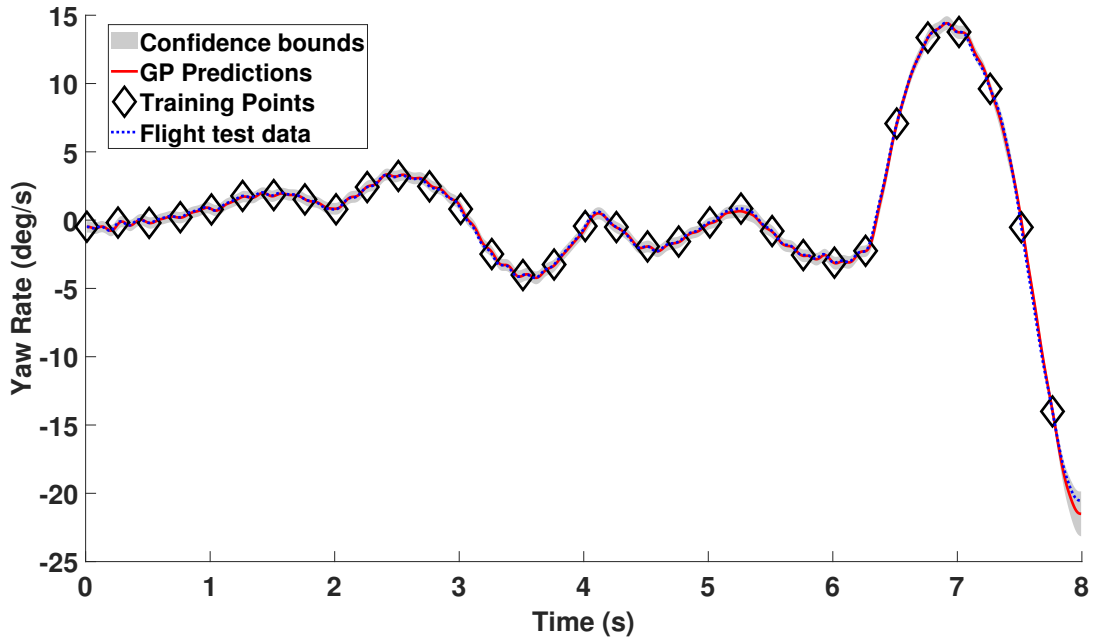


Figure B.30: Gaussian process one step ahead yaw rate predictions using the input structure containing the current pedal pilot input and one lagged yaw rate term using the hyperparameters located by the simulated annealing results shown in Figure B.29.

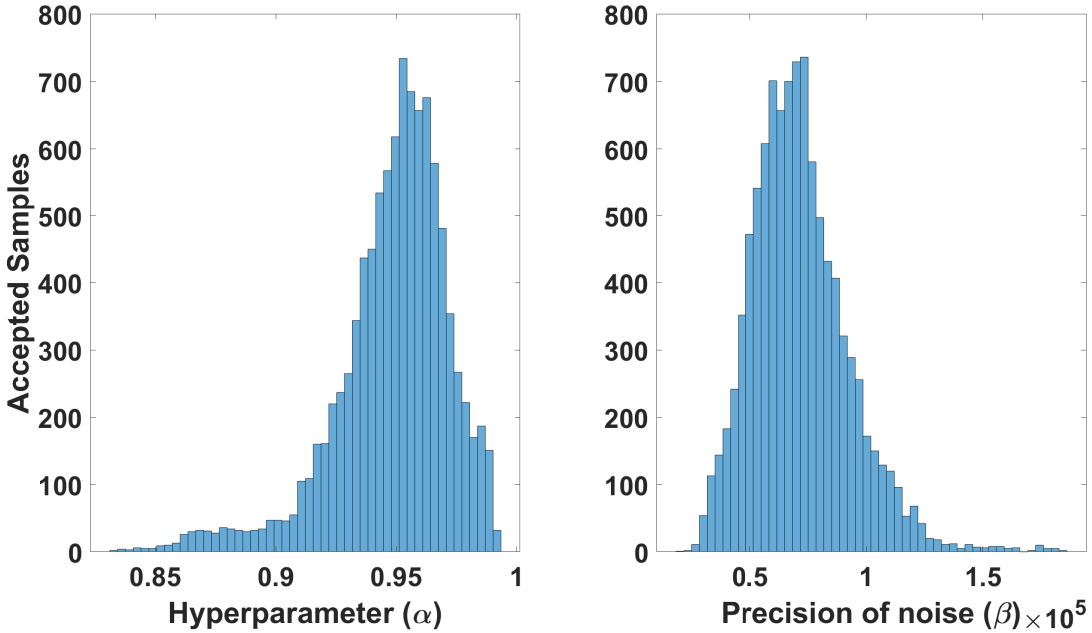


Figure B.31: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing the current pedal pilot input and two lagged yaw rate terms.

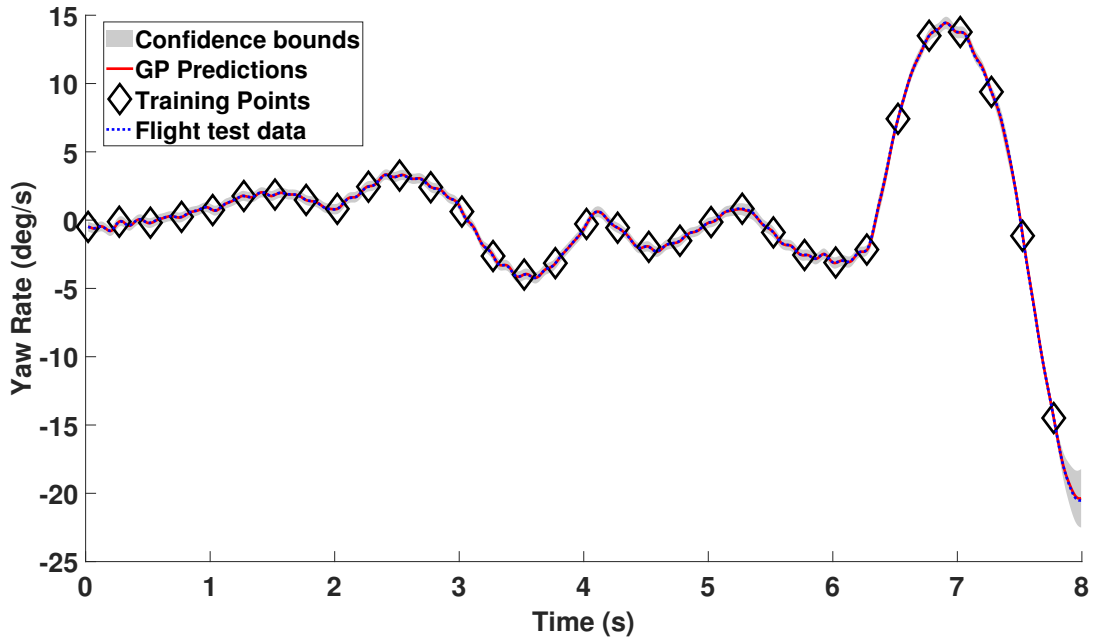


Figure B.32: Gaussian process one step ahead yaw rate predictions using the input structure containing the current pedal pilot input and two lagged yaw rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.31.

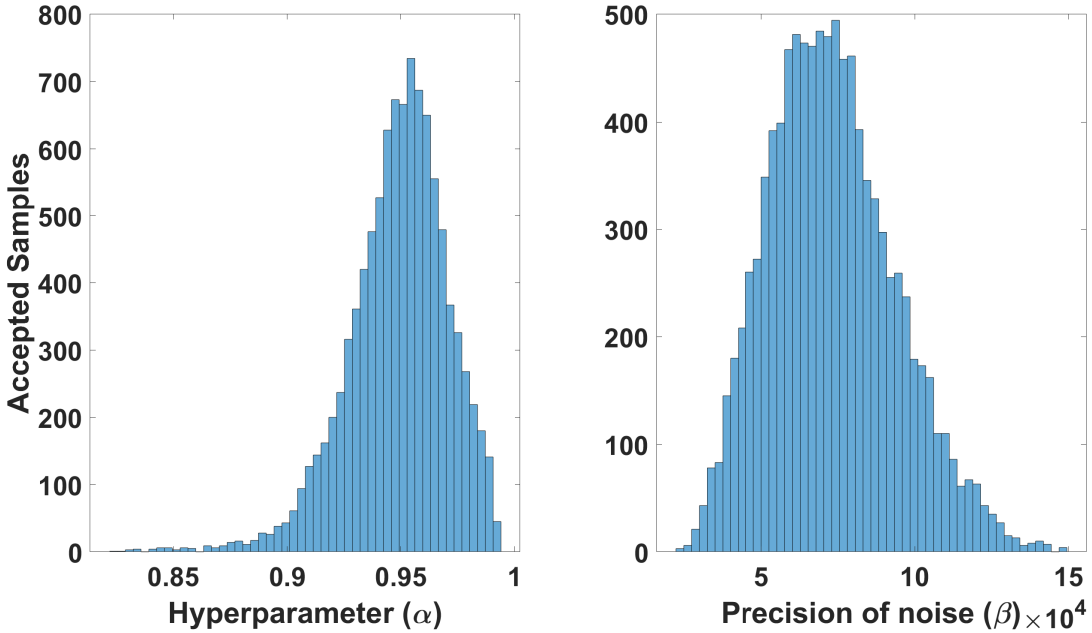


Figure B.33: Simulated annealing results used to create the yaw rate Gaussian process model using the input structure containing the current pedal pilot input and three lagged yaw rate terms.

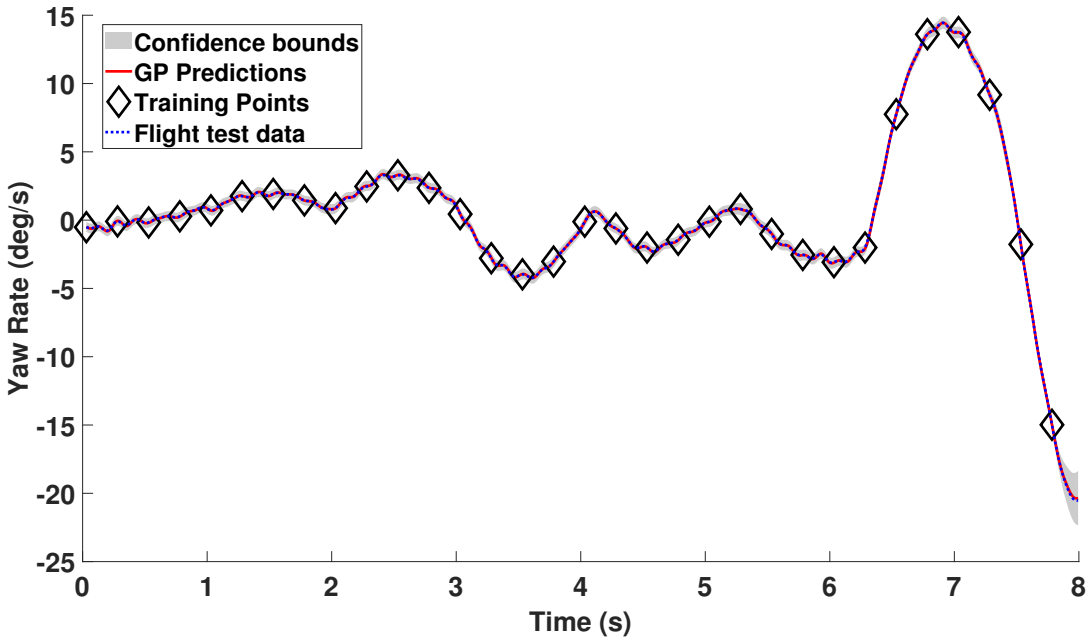


Figure B.34: Gaussian process one step ahead yaw rate predictions using the input structure containing the current pedal pilot input and three lagged yaw rate terms using the hyperparameters located by the simulated annealing results shown in Figure B.33.

B.4 Heave

Figures in this Section relate to the heave response GP results shown in Section 5.1.4.

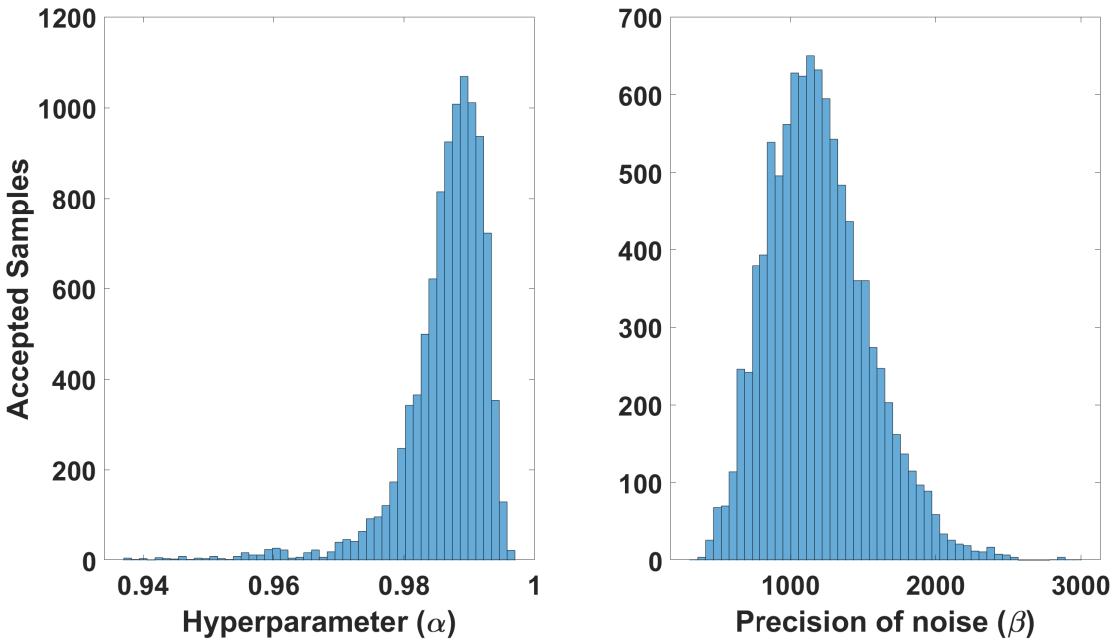


Figure B.35: Simulated annealing results used to create the heave Gaussian process model using the input structure containing all of the current pilot inputs and one lagged heave term.

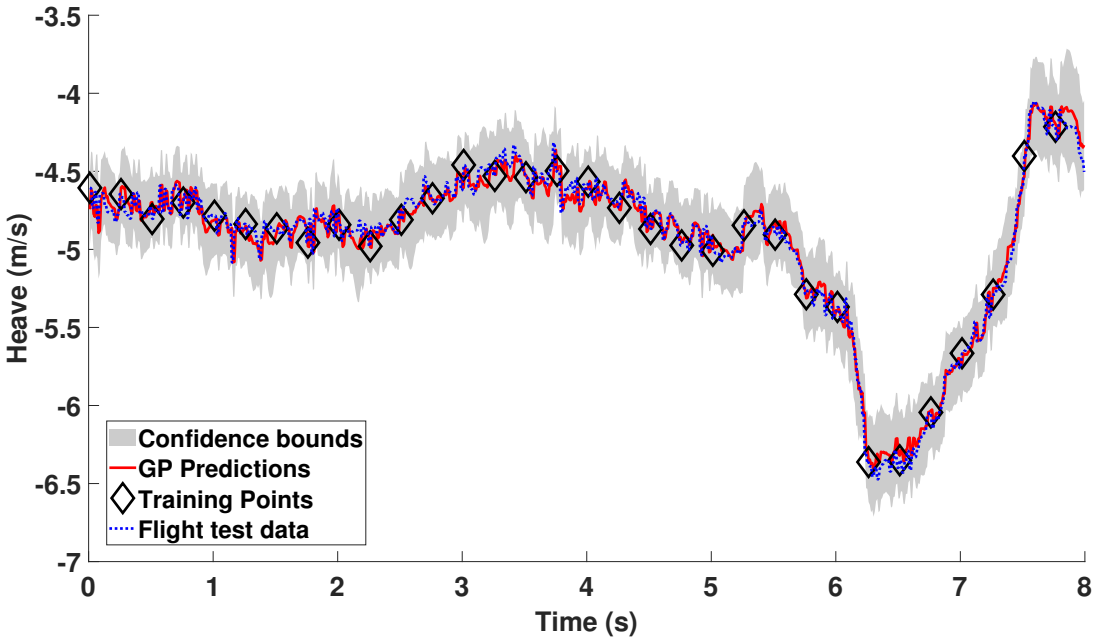


Figure B.36: Gaussian process one step ahead heave predictions using the input structure containing all of the current pilot inputs and one lagged heave term using the hyperparameters located by the simulated annealing results shown in Figure B.35.

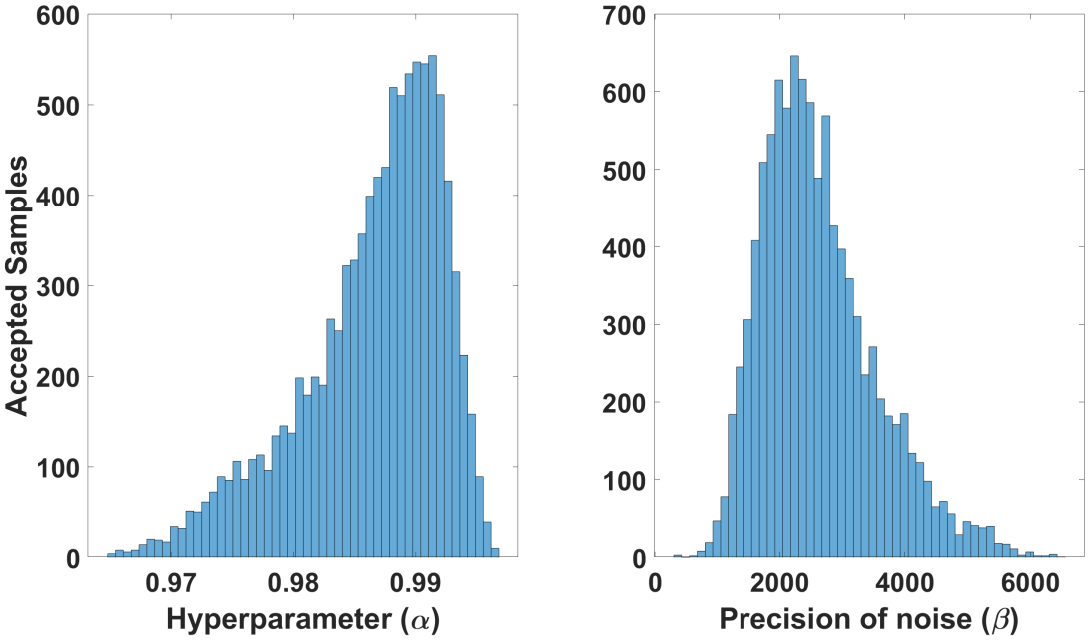


Figure B.37: Simulated annealing results used to create the heave Gaussian process model using the input structure containing all of the current pilot inputs and two lagged heave terms.

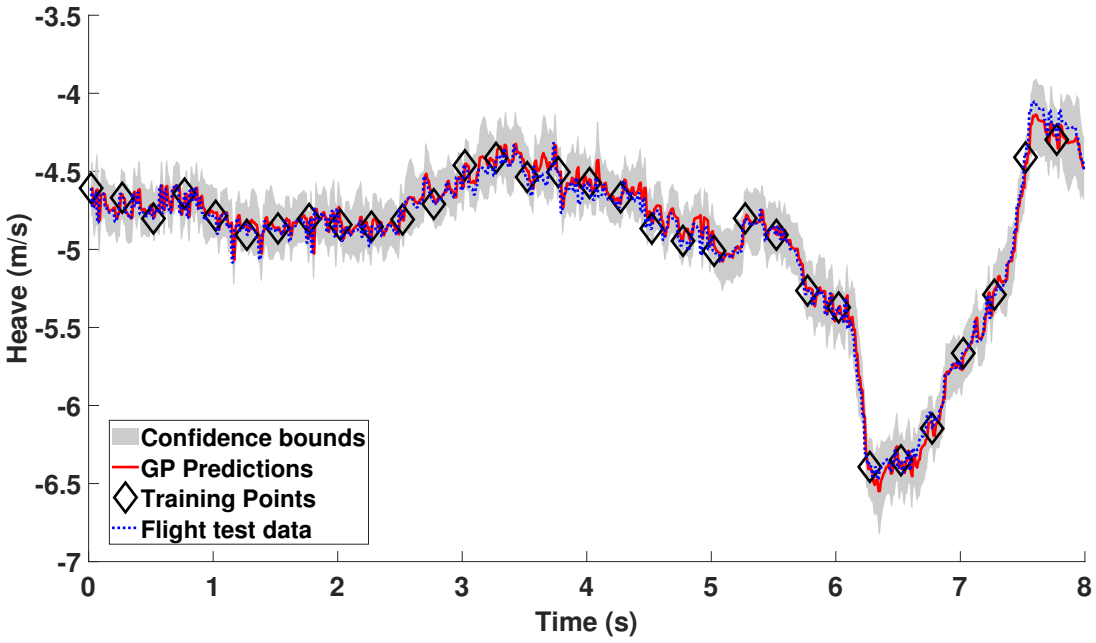


Figure B.38: Gaussian process one step ahead heave predictions using the input structure containing all of the current pilot inputs and two lagged heave terms using the hyperparameters located by the simulated annealing results shown in Figure B.37.

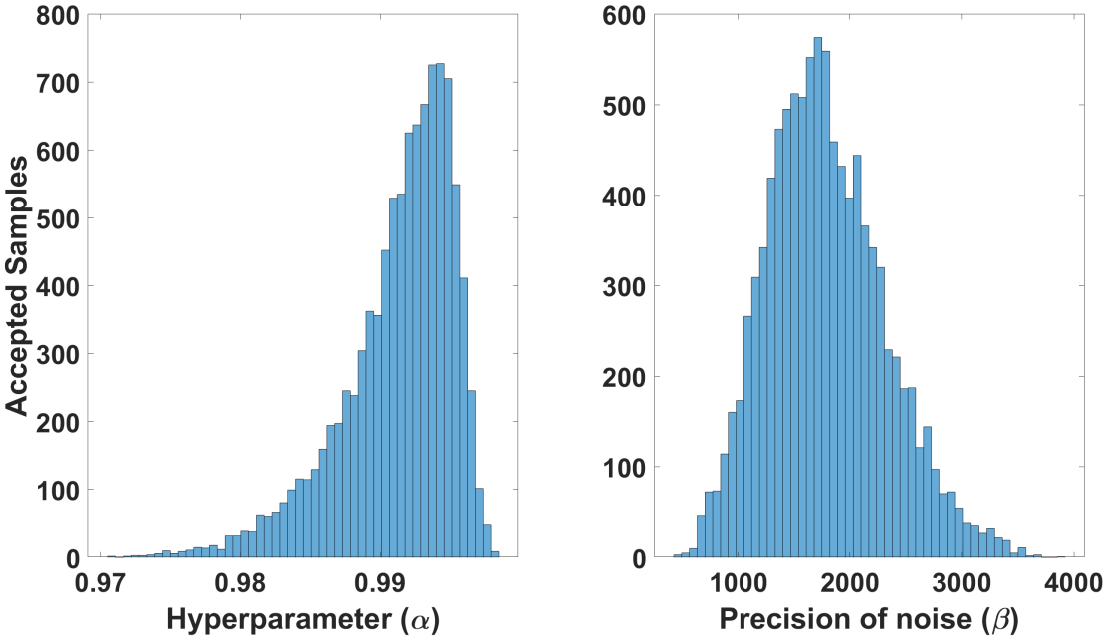


Figure B.39: Simulated annealing results used to create the heave Gaussian process model using the input structure containing all of the current pilot inputs and three lagged heave terms.

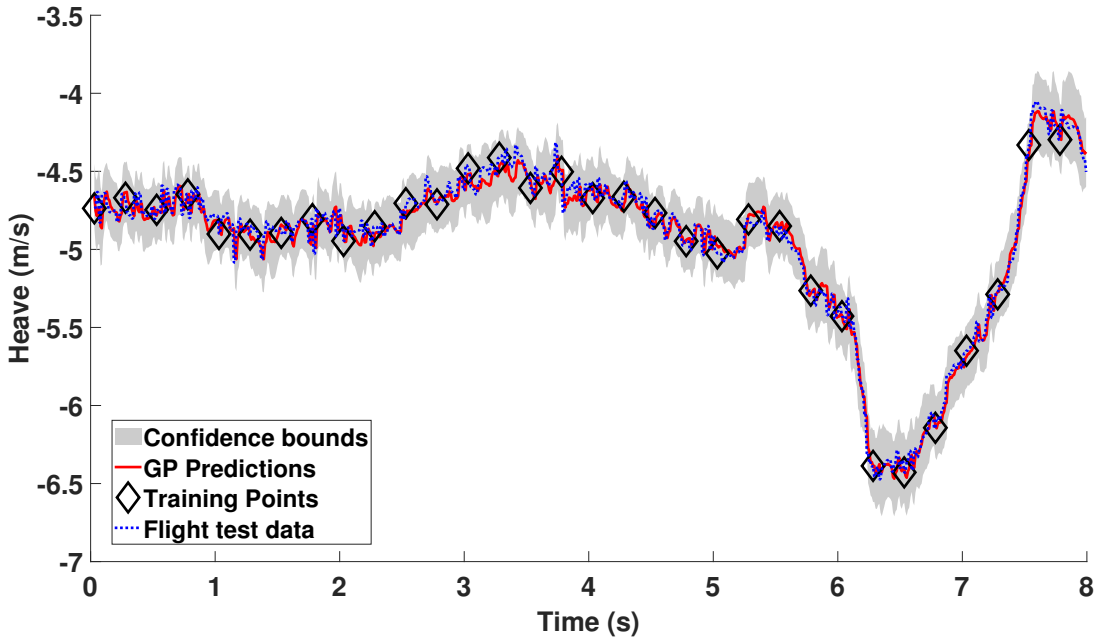


Figure B.40: Gaussian process one step ahead heave predictions using the input structure containing all of the current pilot inputs and three lagged heave terms using the hyperparameters located by the simulated annealing results shown in Figure B.39.

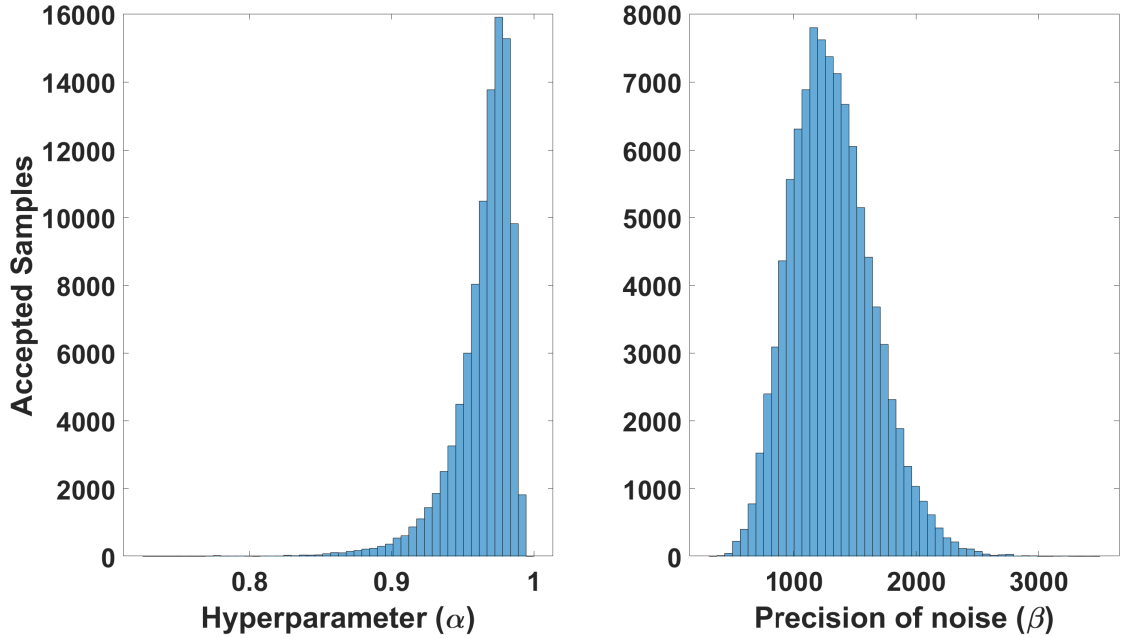


Figure B.41: Simulated annealing results used to create the heave Gaussian process model using the input structure containing the current collective lever pilot input and one lagged heave term.

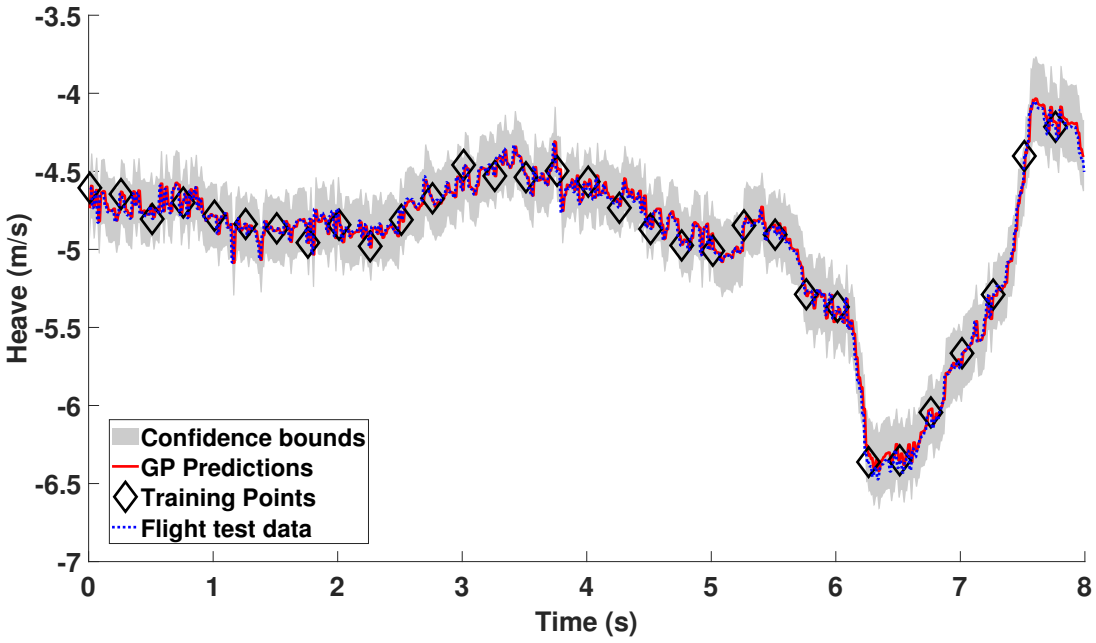


Figure B.42: Gaussian process one step ahead heave predictions using the input structure containing the current collective lever pilot input and one lagged heave term using the hyper-parameters located by the simulated annealing results shown in Figure B.41.

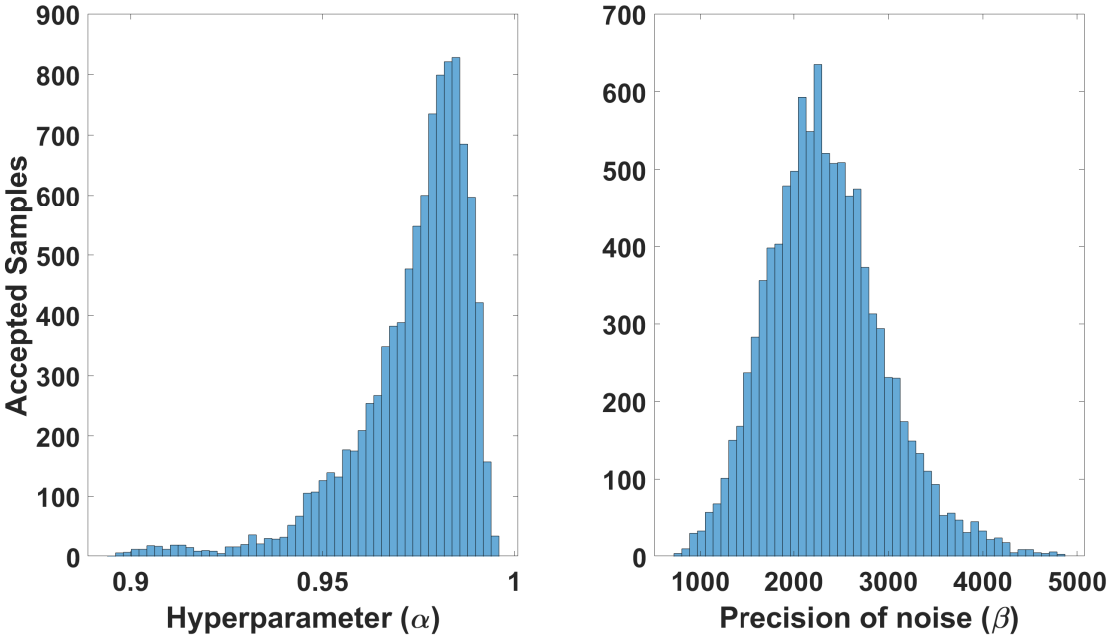


Figure B.43: Simulated annealing results used to create the heave Gaussian process model using the input structure containing the current collective lever pilot input and two lagged heave terms.

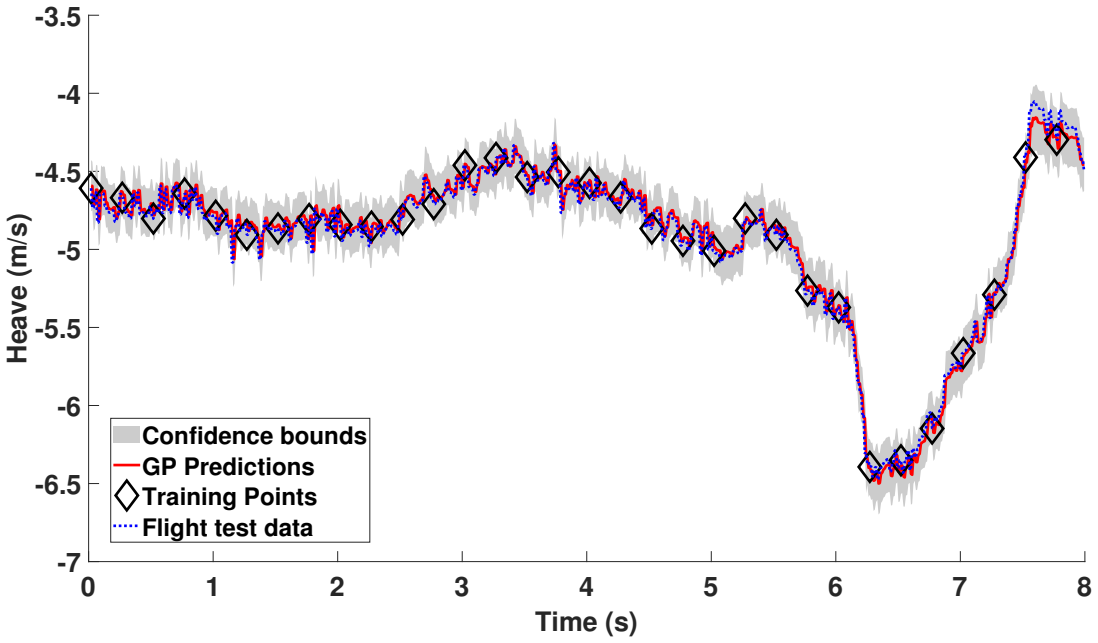


Figure B.44: Gaussian process one step ahead heave predictions using the input structure containing the current collective lever pilot input and two lagged heave terms using the hyper-parameters located by the simulated annealing results shown in Figure B.43.

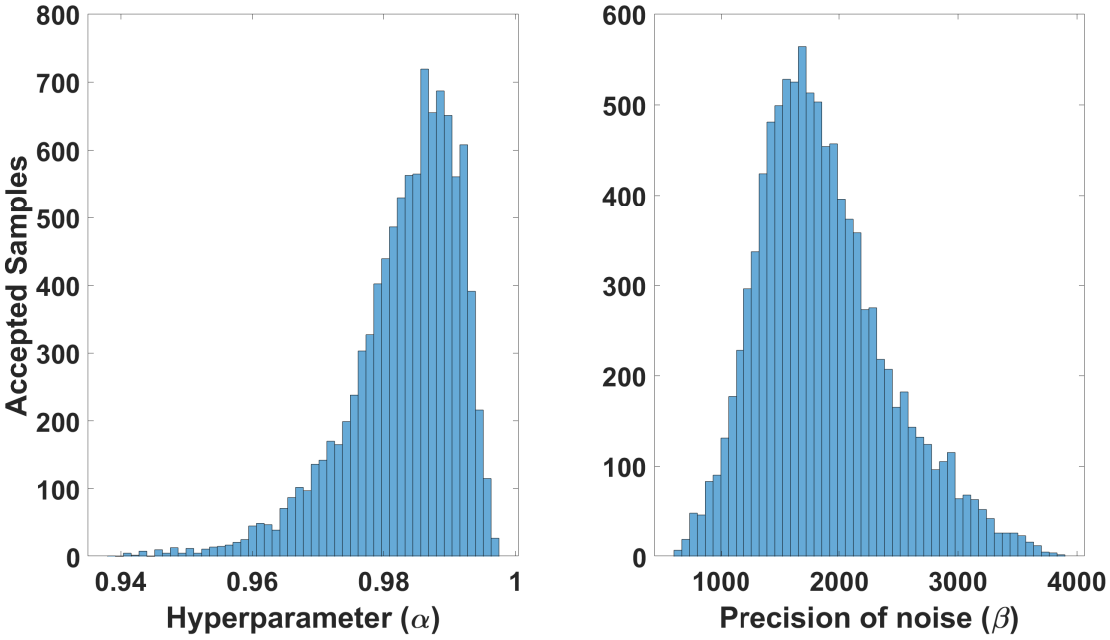


Figure B.45: Simulated annealing results used to create the heave Gaussian process model using the input structure containing the current collective lever pilot input and three lagged heave terms.

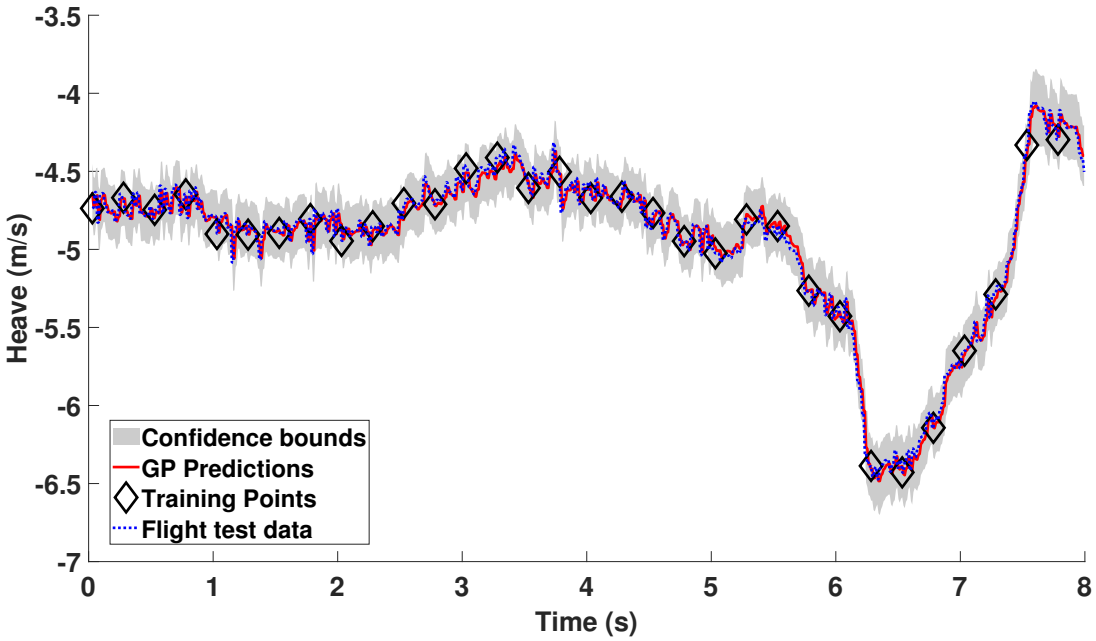


Figure B.46: Gaussian process one step ahead heave predictions using the input structure containing the current collective lever pilot input and three lagged heave terms using the hyperparameters located by the simulated annealing results shown in Figure B.45.

Appendix C

Investigation for GP hyperparameter structures

C.1 Pitch Rate

Figures in this Section relate to the pitch rate response GP results shown in Section 6.1.1.

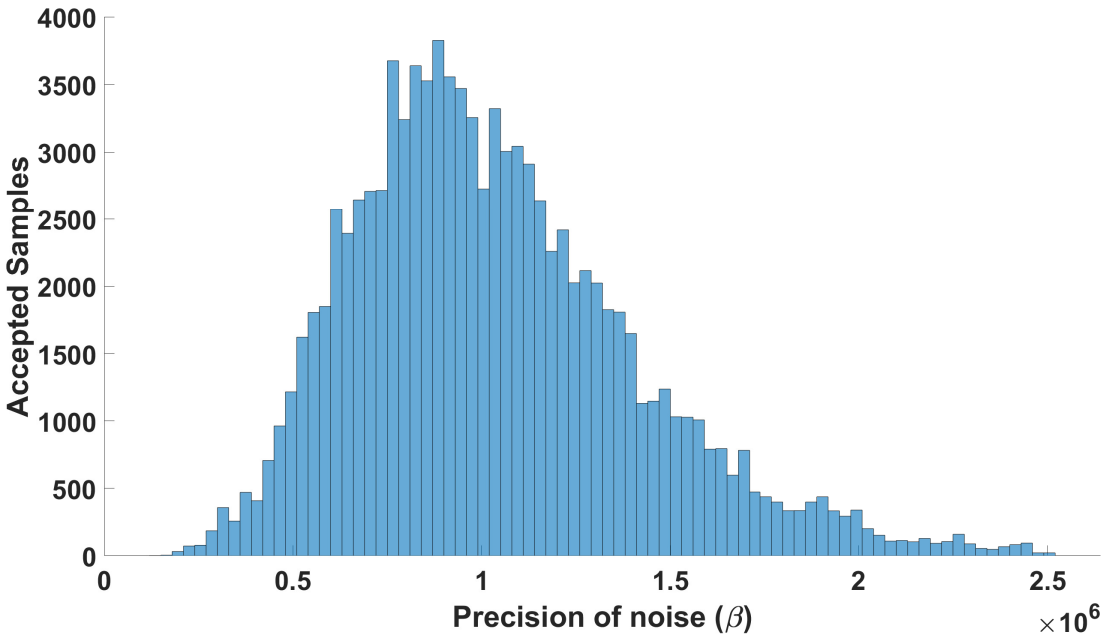


Figure C.1: Simulated annealing results for the precision of noise (β) used to create pitch rate Gaussian process models using the input structure containing all of the current pilot inputs with two lagged pitch rate terms utilising six hyperparameters.

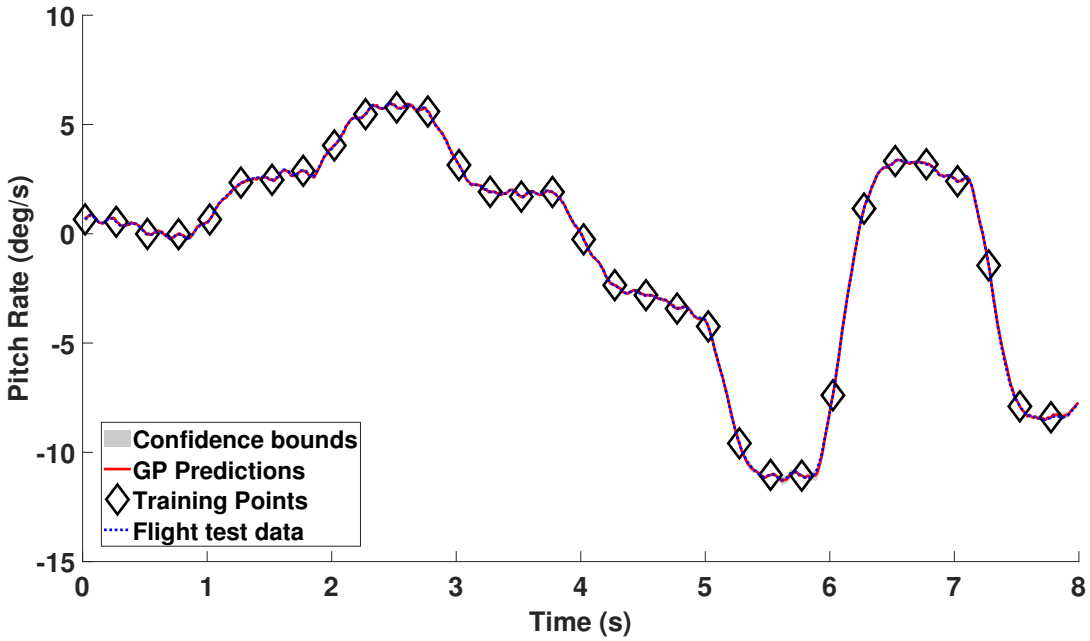


Figure C.2: Gaussian process one step ahead pitch rate predictions using six hyperparameters and the input structure containing all of the current pilot inputs with two lagged pitch rate terms.

C.2 Roll Rate

Figures in this Section relate to the roll rate response GP results shown in Section 6.1.2.

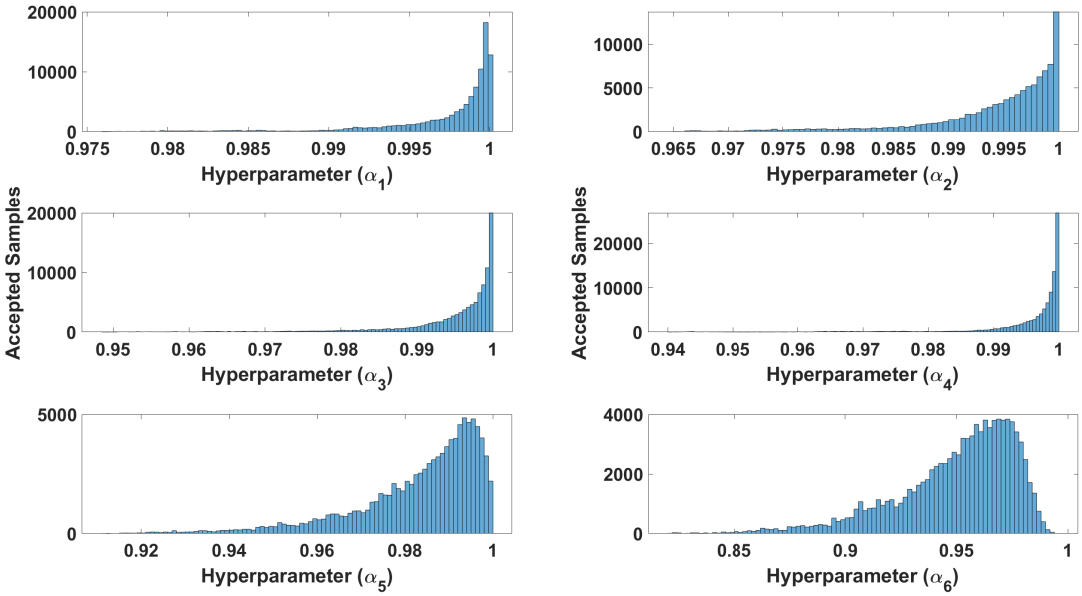


Figure C.3: Simulated annealing results for the hyperparameters used to create roll rate Gaussian process models using the input structure containing all of the current pilot inputs with two lagged roll rate terms utilising six hyperparameters.

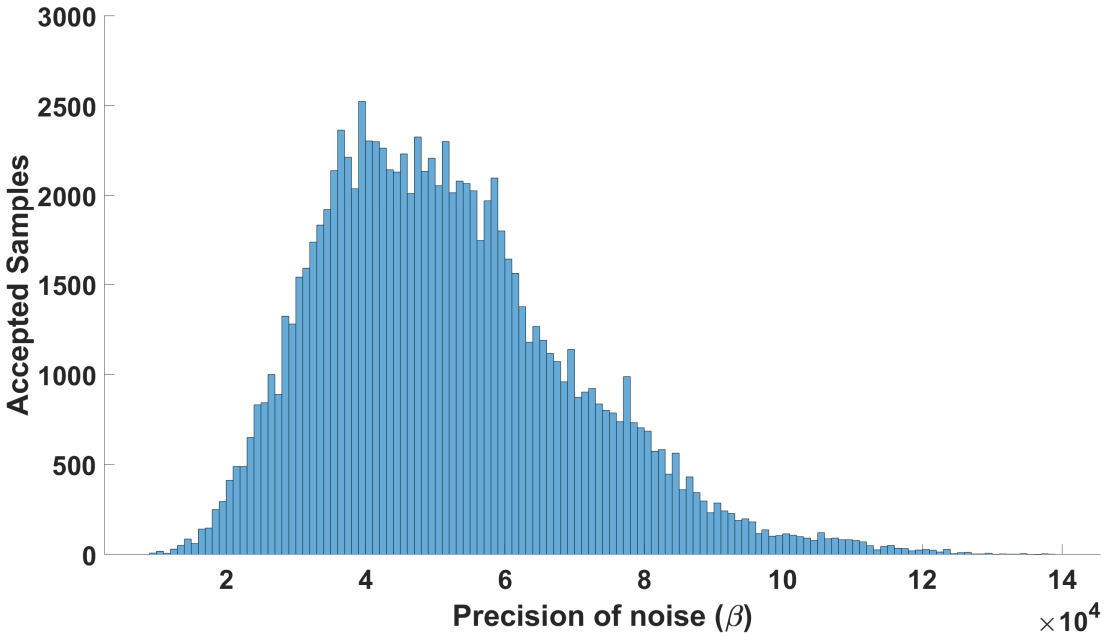


Figure C.4: Simulated annealing results for the precision of noise (β) used to create roll rate Gaussian process models using the input structure containing all of the current pilot inputs with two lagged roll rate terms utilising six hyperparameters.

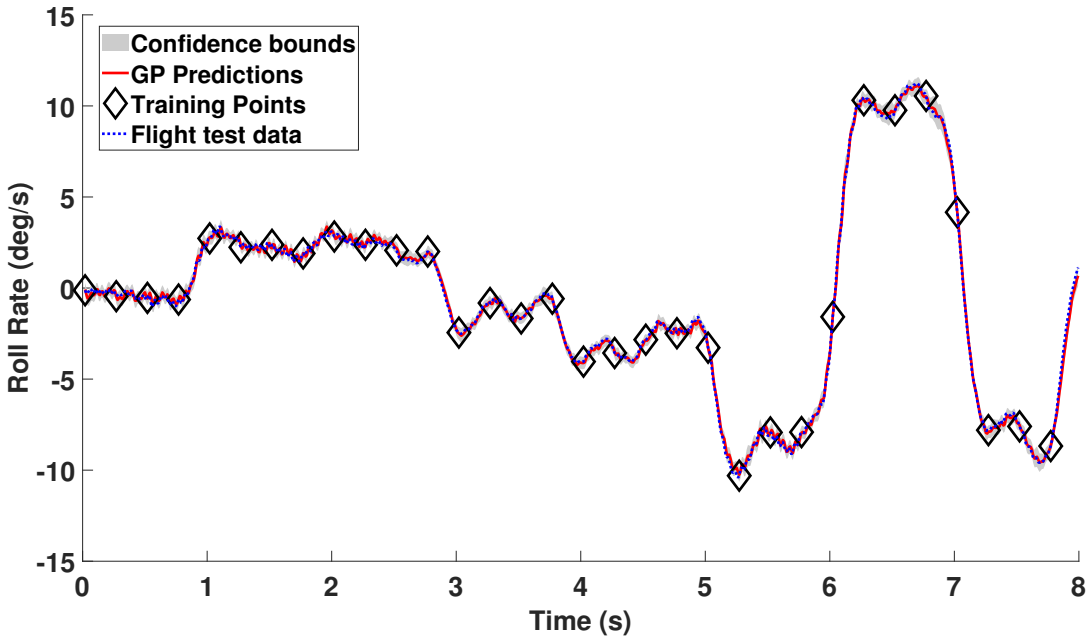


Figure C.5: Gaussian process one step ahead roll rate predictions using six hyperparameters and the input structure containing all of the current pilot inputs with two lagged roll rate terms.

C.3 Yaw Rate

Figures in this Section relate to the yaw rate response GP results shown in Section 6.1.3.

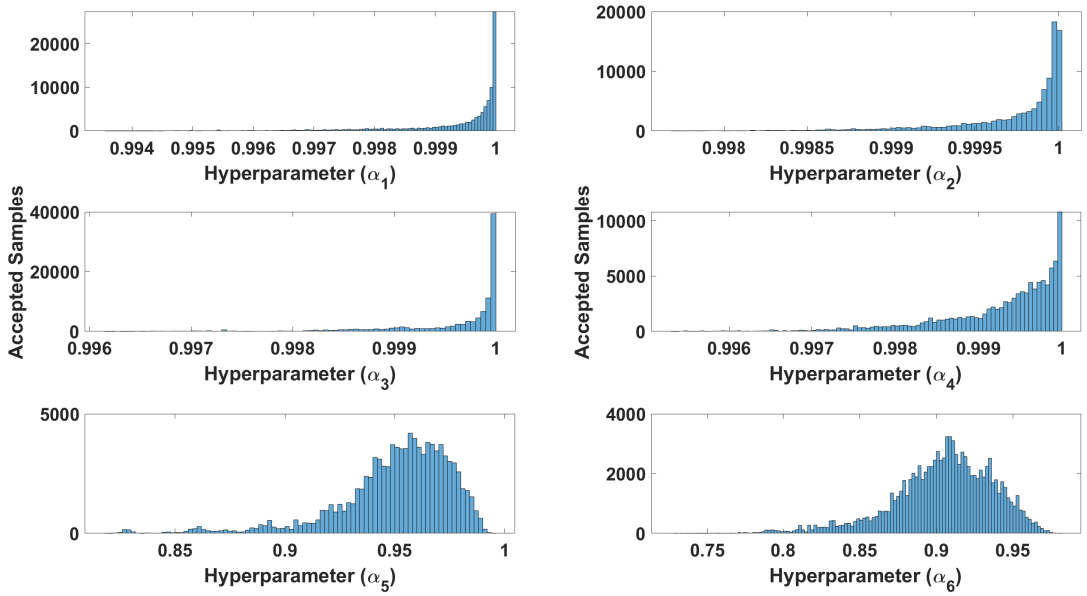


Figure C.6: Simulated annealing results for the hyperparameters used to create yaw rate Gaussian process models using the input structure containing all of the current pilot inputs with two lagged yaw rate terms utilising six hyperparameters.

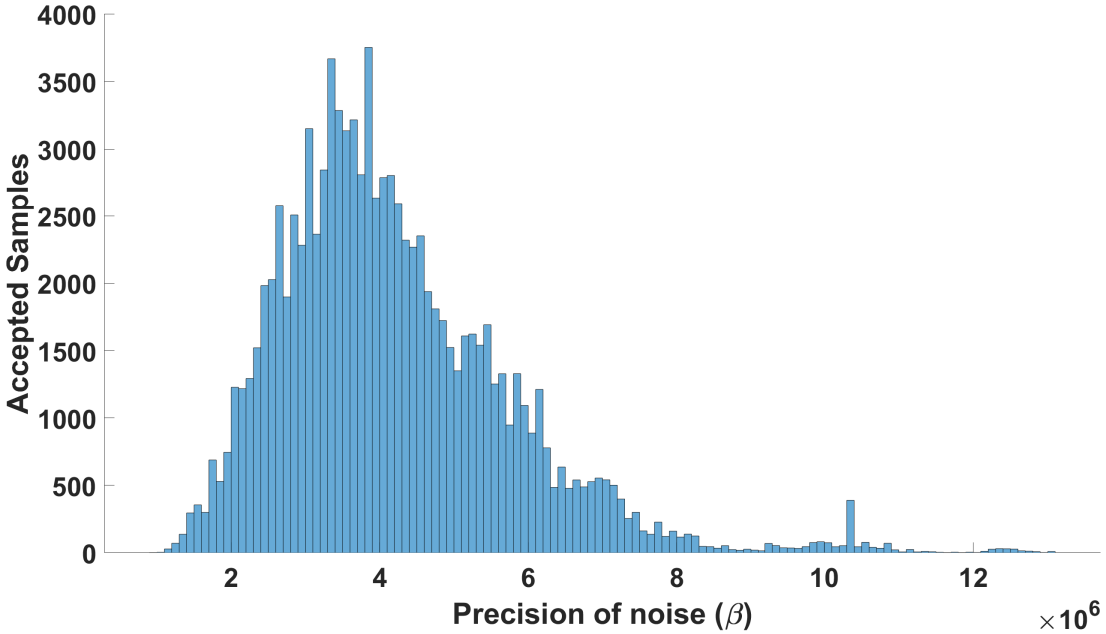


Figure C.7: Simulated annealing results for the precision of noise (β) used to create yaw rate Gaussian process models using the input structure containing all of the current pilot inputs with two lagged yaw rate terms utilising six hyperparameters.

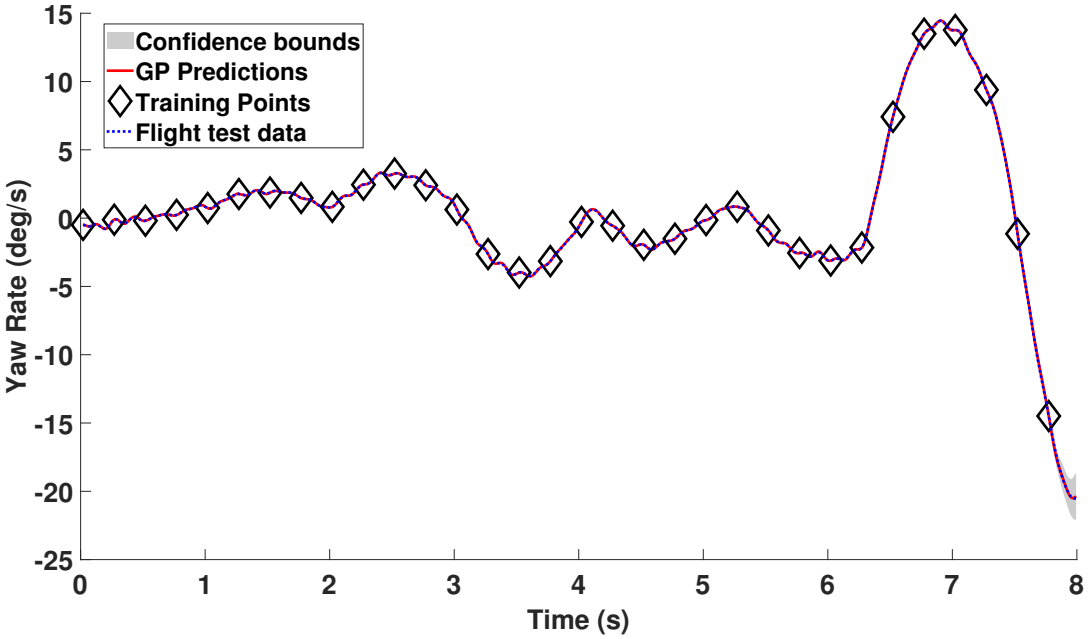


Figure C.8: Gaussian process one step ahead yaw rate predictions using six hyperparameters and the input structure containing all of the current pilot inputs with two lagged yaw rate terms.

C.4 Heave

Figures in this Section relate to the heave response GP results shown in Section 6.1.4.

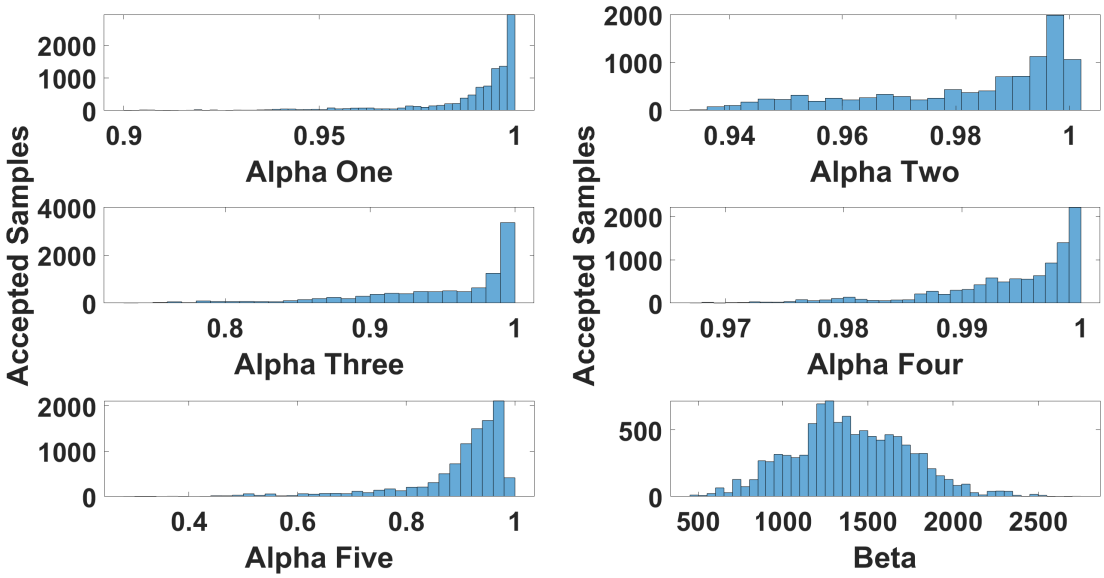


Figure C.9: Simulated annealing results for the hyperparameters used to create heave Gaussian process models using the input structure containing all of the current pilot inputs with one lagged heave term utilising five hyperparameters.

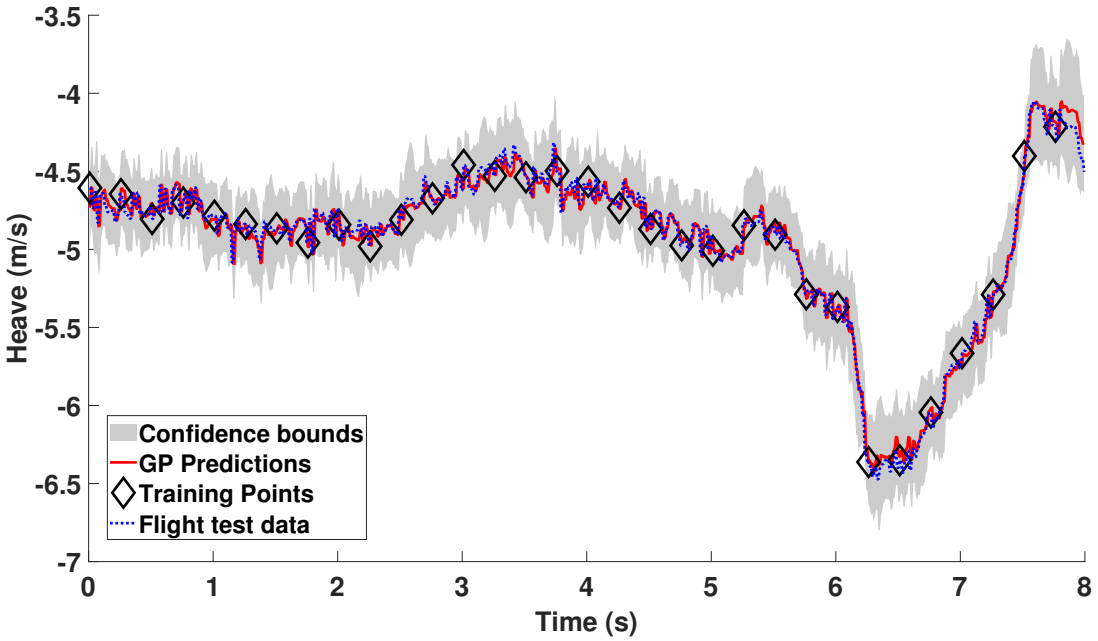


Figure C.10: Gaussian process one step ahead heave predictions using five hyperparameters and the input structure containing all of the current pilot inputs with one lagged heave term.

Appendix D

Lower bound proposed by Hensman et al

To make predictions using a GP requires the inversion of $\boldsymbol{K}_{NN}$, which has a high computational cost, as does the likelihood to find the sparse points. A lower bound of the log-likelihood is therefore required. The lower bound is calculated which will only have an inversion of $\boldsymbol{K}_{SS}$, thus reducing the cost from $O(N^3)$ to $O(NS^2)$. Hensmann et al. used two methods to find the lower bound; the first method used Jensen's inequality to define the lower bound. The Jensen's inequality is shown first, using equation (7.14) it is shown:

$$p(\boldsymbol{y} \mid \boldsymbol{f}_S) = \int p(\boldsymbol{y} \mid \boldsymbol{f}) p(\boldsymbol{f} \mid \boldsymbol{f}_S) d\boldsymbol{f} = \mathbf{E}_{\boldsymbol{f}|\boldsymbol{f}_S}[p(\boldsymbol{y} \mid \boldsymbol{f})]. \tag{D.1}$$

Jensen's inequality is applied to the likelihood, which is:

$$\log(\mathbf{E}_{\boldsymbol{f}|\boldsymbol{f}_S}[p(\boldsymbol{y} \mid \boldsymbol{f})]) \geq \mathbf{E}_{\boldsymbol{f}|\boldsymbol{f}_S}[\log(p(\boldsymbol{y} \mid \boldsymbol{f}))] = \mathcal{L}_1 \tag{D.2}$$

using equation (D.1) it is written as:

$$\log p(\boldsymbol{y} \mid \boldsymbol{f}_S) = \log \left(\int p(\boldsymbol{y} \mid \boldsymbol{f}) p(\boldsymbol{f} \mid \boldsymbol{f}_S) \right) d\boldsymbol{f} \geq \int \log(p(\boldsymbol{y} \mid \boldsymbol{f})) p(\boldsymbol{f} \mid \boldsymbol{f}_S) d\boldsymbol{f} = \mathcal{L}_1. \tag{D.3}$$

One knows $\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{f}) = \prod_{i=1}^{N} p(y_N \mid f_N)$ therefore the lower bound is written as:

$$\mathcal{L}_1 = \sum_{i=1}^{N} \left[\int \log(p(y_i \mid f_i)) p(f_i \mid \boldsymbol{f}_S) df_i \right] \tag{D.4}$$

where:

$$p(y_i \mid f_i) = C \exp\left(-\frac{\beta}{2}(y_i - f_i)^2 \right) \tag{D.5}$$

therefore:

$$\log p(y_i \mid f_i) = \log C - \frac{\beta}{2}(y_i - f_i)^2. \tag{D.6}$$

The lower bound is shown to be:

$$\begin{aligned} \mathcal{L}_1 &= \sum_{i=1}^{N} \int \left[\log C - \frac{\beta}{2}(y_i - f_i)^2 \right] \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i \\ &= \sum_{i=1}^{N} \left[\underbrace{\log C \int \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i}_{\text{Part A}} - \underbrace{\frac{\beta}{2} \int (y_i - f_i)^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i}_{\text{Part B}} \right] \end{aligned} \tag{D.7}$$

where Σ_{ii} are the elements of $\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma} = \boldsymbol{K}_{NN} - \boldsymbol{Q}$, and where $\boldsymbol{Q} = \boldsymbol{K}_{NS}^T \boldsymbol{K}_{SS}^{-1} \boldsymbol{K}_{NS}$. Using equation (D.7) focusing on part A only, one can state:

$$\log C \int \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i = \log C \tag{D.8}$$

therefore equation (D.7) is written as:

$$\mathcal{L}_1 = \sum_{i=1}^{N} \left[\log C - \frac{\beta}{2} \int (y_i - f_i)^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) \right] df_i. \tag{D.9}$$

One can now focus on part B by expanding the brackets and splitting the terms up into three parts:

$$\mathcal{L}_1 = \sum_{i=1}^{N} \left[\log C - \frac{\beta}{2} \int \left(\underbrace{f_i^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii})}_{\text{Part C}} - \underbrace{2 f_i y_i \mathcal{N}(f_i \mid \lambda_n, \Sigma_{ii})}_{\text{Part D}} + \underbrace{y_i^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii})}_{\text{Part E}} \right) df_i \right]. \tag{D.10}$$

Using the proof shown in Appendices F.1 and Ap: Part Three, one can solve part C, D and E, these are shown as:

$$\text{Part C} = \int f_i^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i = \Sigma k_{ii} + \lambda_i^2$$

$$\text{Part D} = \int 2 f_i y_i \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i = 2 y_i \lambda_i$$

$$\text{Part E} = \int y_i^2 \mathcal{N}(f_i \mid \lambda_i, \Sigma_{ii}) df_i = y_i^2.$$

The lower bound is written as:

$$\begin{aligned} \mathcal{L}_1 &= \sum_{i=1}^{N} \left[\log C - \frac{\beta}{2} \left(\Sigma_{ii} + \lambda_i^2 - 2 y_i \lambda_i + y_i^2 \right) \right] \\ &= \sum_{i=1}^{N} \left[\log C - \frac{\beta}{2} \left(\Sigma_{ii} + (y_i - \lambda_i)^2 \right) \right]. \end{aligned} \tag{D.11}$$

Alternatively, one can show the lower bound to be:

$$\exp(\mathcal{L}_1) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \lambda_i, \beta^{-1}) \exp\left(-\frac{\beta}{2} \Sigma_{ii} \right) \tag{D.12}$$

or more generally:

$$\exp(\mathcal{L}_1) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \boldsymbol{I}\beta^{-1}) \exp\left(-\frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q}_{NN}) \right) \tag{D.13}$$

The $Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q})$ is found without having to fully evaluate $(\boldsymbol{K}_{NN} - \boldsymbol{Q})$, this reducing the complexity to $O(NS^2)$. The other method to find the same lower bound is by using the KL divergence. The KL divergence measures the 'closeness' of the two probability density functions (pdf). This method is not needed to find the lower bound; however, it does cement the proof. One can state the KL divergence between $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$ and $p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})$ as:

$$KL\left(p(\boldsymbol{f} \mid \boldsymbol{f}_S) \mid\mid p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y}) \right) = \int p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log\left(\frac{p(\boldsymbol{f} \mid \boldsymbol{f}_S)}{p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})} \right) d\boldsymbol{f}. \tag{D.14}$$

The aim is to find the difference between the likelihood $p(\boldsymbol{y} \mid \boldsymbol{u})$ and a lower bound. This is achieved by using Bayes law, one can state:

$$p(\boldsymbol{f} \mid \boldsymbol{f}_S) = \frac{p(\boldsymbol{f}, \boldsymbol{f}_S)}{p(\boldsymbol{f}_S)}, \;\; p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y}) = \frac{p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y})}{p(\boldsymbol{f}_S \mid \boldsymbol{y})} \tag{D.15}$$

therefore:

$$\frac{p(\boldsymbol{f} \mid \boldsymbol{f}_S)}{p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})} = \frac{p(\boldsymbol{f}, \boldsymbol{f}_S)p(\boldsymbol{f}_S \mid \boldsymbol{y})}{p(\boldsymbol{f}_S)p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y})} = \frac{p(\boldsymbol{f}, \boldsymbol{f}_S)p(\boldsymbol{y} \mid \boldsymbol{f}_S)}{p(\boldsymbol{y})p(\boldsymbol{f}, \boldsymbol{f}_S \mid \boldsymbol{y})} = \frac{p(\boldsymbol{y} \mid \boldsymbol{f}_S)}{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S)} \tag{D.16}$$

applying logarithm to both sides:

$$\log\left(\tfrac{p(\boldsymbol{f}|\boldsymbol{u})}{p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{y})}\right) = \log\left(\tfrac{p(\boldsymbol{y}|\boldsymbol{f}_S)}{p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{f}_S)}\right) = \log p(\boldsymbol{y} \mid \boldsymbol{f}_S) - \log p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S). \tag{D.17}$$

Referring to equation (D.16) and (D.15) the KL divergence is written as:

$$KL\left(p(\boldsymbol{f} \mid \boldsymbol{f}_S) \mid\mid p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})\right) = \int p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log\left(\tfrac{p(\boldsymbol{y}|\boldsymbol{f}_S)}{p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{f}_S)}\right) d\boldsymbol{f} \tag{D.18}$$

using equation (D.17) one can write:

$$KL\left(p(\boldsymbol{f} \mid \boldsymbol{f}_S) \mid\mid p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})\right) = \int p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log p(\boldsymbol{y} \mid \boldsymbol{f}_S) d\boldsymbol{f} - \int p(\boldsymbol{f} \mid \boldsymbol{f}_S) p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S) d\boldsymbol{f}. \tag{D.19}$$

Knowing $\int (\boldsymbol{f} \mid \boldsymbol{u}) df = 1$, the KL divergence is shown to be:

$$KL\left(p(\boldsymbol{f} \mid \boldsymbol{f}_S) \mid\mid p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})\right) = \log p(\boldsymbol{y} \mid \boldsymbol{f}_S) - \int p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S) d\boldsymbol{f} \tag{D.20}$$

where:

$$L_1 = \int p(\boldsymbol{f} \mid \boldsymbol{f}_S) \log p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}_S) d\boldsymbol{f}.$$

The difference between $\log p(\boldsymbol{y} \mid \boldsymbol{f}_S)$ and the lower bound is the KL divergence between $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$ and $p(\boldsymbol{f} \mid \boldsymbol{f}_S, \boldsymbol{y})$. This is the cost function for optimisation; the difference between $\log p(\boldsymbol{y} \mid \boldsymbol{u})$

is minimised. The lower bound is written as using equation (D.13) :

$$\begin{aligned}\mathcal{L}_1 &= \log\left(\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \boldsymbol{I}\beta^{-1}) \exp\left(-\frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q})\right)\right) \\ &= \log\left(\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \boldsymbol{I}\beta^{-1}) - \frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q})\right)\end{aligned} \tag{D.21}$$

To be able to use the lower bound as a cost function, it has been written in terms of variables rather than in terms of distributions. To do this, the multivariate Gaussian is required, which shows the lower bound to be:

$$\begin{aligned}\mathcal{L}_1 &= \log\left(\frac{1}{2\pi^{\frac{N}{2}} \mid \boldsymbol{I}\beta^{-1} \mid^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\lambda})^T (\boldsymbol{I}\beta^{-1})^{-1} (\boldsymbol{y} - \boldsymbol{\lambda})\right)\right) - \frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q}) \\ &= \log\left(\frac{1}{2\pi^{\frac{N}{2}} \mid \boldsymbol{I}\beta^{-1|^{\frac{1}{2}}}}\right) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\lambda})^T (\boldsymbol{I}\beta^{-1})^{-1} (\boldsymbol{y} - \boldsymbol{\lambda}) - \frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q}) \\ &= \log(1) - \log(2\pi^{\frac{N}{2}} \mid \boldsymbol{I}\beta^{-1} \mid^{\frac{1}{2}}) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\lambda})^T (\boldsymbol{I}\beta^{-1})^{-1} (\boldsymbol{y} - \boldsymbol{\lambda}) - \frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q}) \\ &= -\log(2\pi^{\frac{N}{2}}) - \log(\mid \boldsymbol{I}\beta^{-1} \mid^{\frac{1}{2}}) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\lambda})^T (\boldsymbol{I}\beta^{-1})^{-1} (\boldsymbol{y} - \boldsymbol{\lambda}) - \frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q}) \\ &= -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log(\mid \boldsymbol{I}\beta^{-1} \mid) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\lambda})^T (\boldsymbol{I}\beta^{-1})^{-1} (\boldsymbol{y} - \boldsymbol{\lambda}) - \frac{\beta}{2} Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q}) \\ &= -\frac{1}{2}\left(N\log(2\pi) + \log(\mid \boldsymbol{I}\beta^{-1} \mid) + (\boldsymbol{y} - \boldsymbol{\lambda})^T (\boldsymbol{I}\beta^{-1})(\boldsymbol{y} - \boldsymbol{\lambda}) + \beta Tr(\boldsymbol{K}_{NN} - \boldsymbol{Q})\right)\end{aligned} \tag{D.22}$$

Appendix E

Covariance of the posterior for the sparse GP method

The properties of Gaussian's state that:

$$p(\boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{A}^{-1}) \quad p(\boldsymbol{x}_b \mid \boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{H}\boldsymbol{x}_a + \boldsymbol{b}, \boldsymbol{L}^{-1})$$

therefore the covariance and mean is given by:

$$cov[\boldsymbol{x}_a \mid \boldsymbol{x}_b] = (\boldsymbol{A} + \boldsymbol{H}^T\boldsymbol{L}\boldsymbol{H})^{-1}$$

$$E[\boldsymbol{x}_a \mid \boldsymbol{x}_b] = (\boldsymbol{A} + \boldsymbol{H}^T\boldsymbol{L}\boldsymbol{H})^{-1}(\boldsymbol{A}\boldsymbol{u} + \boldsymbol{H}^T\boldsymbol{L}(\boldsymbol{x}_b - \boldsymbol{b})).$$

Comparing the above properties with $p(\boldsymbol{f}_S, \mid \boldsymbol{y})$ it is seen that:

$$\boldsymbol{A}^{-1} = \boldsymbol{K}_{SS}, \boldsymbol{\mu} = 0, \boldsymbol{x}_a = \boldsymbol{f}_S, \boldsymbol{x}_b = \boldsymbol{y}, \boldsymbol{H} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}, \boldsymbol{L}^{-1} = \boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T + \boldsymbol{I}\beta^{-1}.$$

The matrix inverse lemma is used to show the inverse involved in the covariance, which states:

$$(\boldsymbol{A} + \boldsymbol{H}^T\boldsymbol{L}\boldsymbol{H})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{H}^T(\boldsymbol{L}^{-1} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{A}^{-1}$$

therefore:

$$\begin{aligned}
\mathrm{Cov}[\boldsymbol{f}_S, \mid \boldsymbol{y}] &= \boldsymbol{K}_{SS} - \boldsymbol{K}_{SS}\boldsymbol{K}_{NS}^T\boldsymbol{K}_{SS}^{-1}(\boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T + \boldsymbol{I}\beta^{-1} + \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T)^{-1}\boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{SS} \\
&= \boldsymbol{K}_{SS} - \boldsymbol{K}_{NS}^T(\boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T + \boldsymbol{I}\beta^{-1} + \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^T)^{-1}\boldsymbol{K}_{NS} \\
&= \boldsymbol{K}_{SS} - \boldsymbol{K}_{NS}^T(\boldsymbol{K}_{NN} + \boldsymbol{I}\beta)^{-1}\boldsymbol{K}_{NS} \\
&= \boldsymbol{K}_{SS} - \boldsymbol{K}_{NS}^T\boldsymbol{C}^{-1}\boldsymbol{K}_{NS}
\end{aligned}$$

where:

$$\boldsymbol{C} = \boldsymbol{K}_{NN} + \boldsymbol{I}\beta^{-1}.$$

Appendix F

Solving for the integral one in section 7.2.2

F.1 Part Two

One wants to find $\int \boldsymbol{f} p(\boldsymbol{f} \mid \boldsymbol{f}_S) d\boldsymbol{f}$. Stating:

$$\mathrm{E}[x] = \int_{-\infty}^{\infty} (x \mid \mu, V)\, x dx = \mu$$

writing $p(\boldsymbol{f} \mid \boldsymbol{f}_S)$ as:

$$p(\boldsymbol{f} \mid \boldsymbol{f}_S) = \mathcal{N}\left(\boldsymbol{f} \mid \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S, \boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^{T}\right)$$

therefore:

$$\mathrm{E}[\boldsymbol{f}] = \int_{-\infty}^{\infty} \left(\boldsymbol{f} \mid \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S, \boldsymbol{K}_{NN} - \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^{T}\right) \boldsymbol{f} d\boldsymbol{f} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S.$$

Writing the final solution for $\int \boldsymbol{f} p(\boldsymbol{f} \mid \boldsymbol{f}_S) d\boldsymbol{f}$:

$$\int \boldsymbol{f} p(\boldsymbol{f} \mid \boldsymbol{f}_S) d\boldsymbol{f} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S$$

where $\boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S = \boldsymbol{\lambda}$

F.2 Part Three

One wants to find $\int \boldsymbol{f}\boldsymbol{f}^T\mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\lambda}, \boldsymbol{\Sigma})$, which can be written as:

$$\sum_{i=1}^{N} \int f_i^2 \mathcal{N}(f_i \mid \lambda_i.\Sigma_{ii}) df_i.$$

One knows:

$$\mathrm{E}[(x-\mu)^2] = V$$

$$\mathrm{E}[x^2] - 2\mu\mathrm{E}[x] + \mu^2 = V$$

$\mathrm{E}[x]$ is equal to the mean μ, therefore:

$$\mathrm{E}[x^2] - \mu^2 = V$$

$$\mathrm{E}[x^2] = V + \mu^2.$$

The integral is written as:

$$x^2\mathcal{N}(x \mid \mu, V)dx = V + \mu^2$$

therefore:

$$\sum_{i=1}^{N} \int f_i^2 \mathcal{N}(f_i \mid \lambda_i.\Sigma_{ii}) df_i = \sum_{i=1}^{N} \Sigma_{ii} + \lambda_i^2$$

Appendix G

Using properties of Gaussian to solve integral two

Defining integral two (Int2) as:

$$\text{Int2} = \int \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{f}_S, \beta^{-1}\boldsymbol{I})\mathcal{N}(\boldsymbol{f}_S \mid \boldsymbol{0}, \boldsymbol{K}_{SS})d\boldsymbol{f}_S$$

using Bayes' one can show:

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{f}_S)p(\boldsymbol{f}_S)d\boldsymbol{f}_S.$$

It is therefore possible to use properties of Gaussian which states:

$$p(\boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{x}_a \mid \boldsymbol{\mu}, \boldsymbol{A}^{-1}) \quad p(\boldsymbol{x}_b \mid \boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{x}_b \mid \boldsymbol{H}\boldsymbol{x}_a + \boldsymbol{b}, \boldsymbol{L}^{-1})$$

where the marginal distribution is given by:

$$p(x_b) = \mathcal{N}(\boldsymbol{x}_b \mid \boldsymbol{H}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{H}\boldsymbol{A}^{-1}\boldsymbol{H}^T).$$

Comparing to Int2 the variables are:

$$\boldsymbol{x}_a = \boldsymbol{f}_S, \;\; \boldsymbol{\mu} = \boldsymbol{0}, \;\; \boldsymbol{A}^{-1} = \boldsymbol{K}_{SS}, \;\; \boldsymbol{x}_b = \boldsymbol{y}, \;\; \boldsymbol{H} = \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}, \;\; \boldsymbol{b} = \boldsymbol{0}, \;\; \boldsymbol{L}^{-1} = \beta^{-1}\boldsymbol{I}$$

therefore Int2 is:

$$\begin{aligned}\text{Int2} &= \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1} \times \mathbf{0} + \mathbf{0}, \beta^{-1}\boldsymbol{I} + \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{SS}\boldsymbol{K}_{NS}^{T}\boldsymbol{K}_{SS}^{-1}) \\ &= \mathcal{N}(\boldsymbol{y} \mid \mathbf{0}, \beta^{-1}\boldsymbol{I} + \boldsymbol{K}_{NS}\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{NS}^{T})\end{aligned}$$

Appendix H

Sparse GP Results

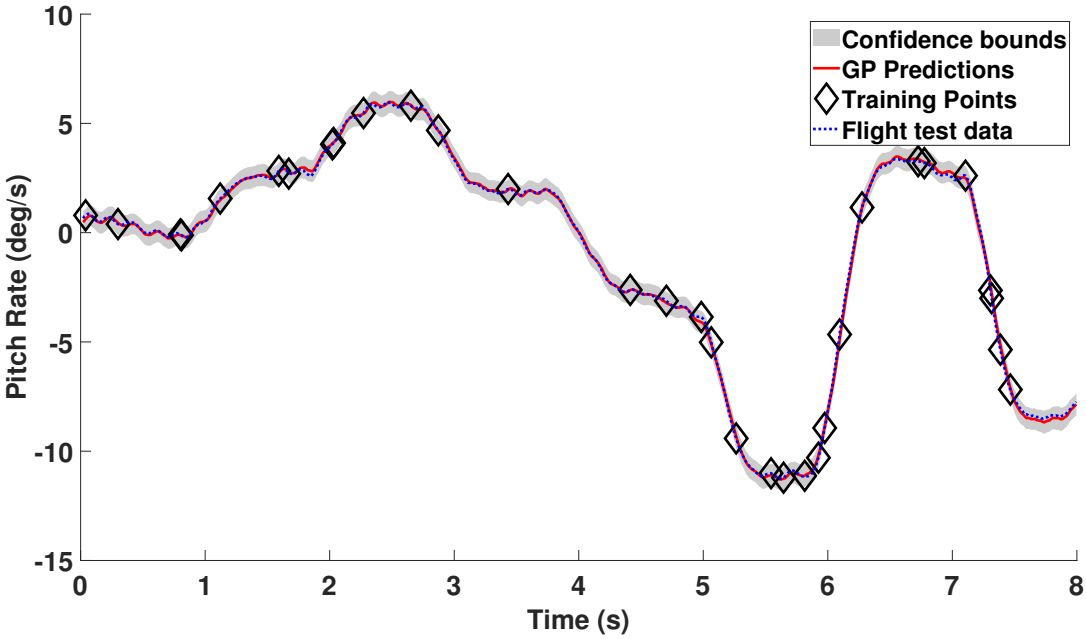


Figure H.1: Sparse Gaussian process one step ahead pitch rate prediction using 32 training points and one hyperparameter (α) (Case 1 in Table 8.1).

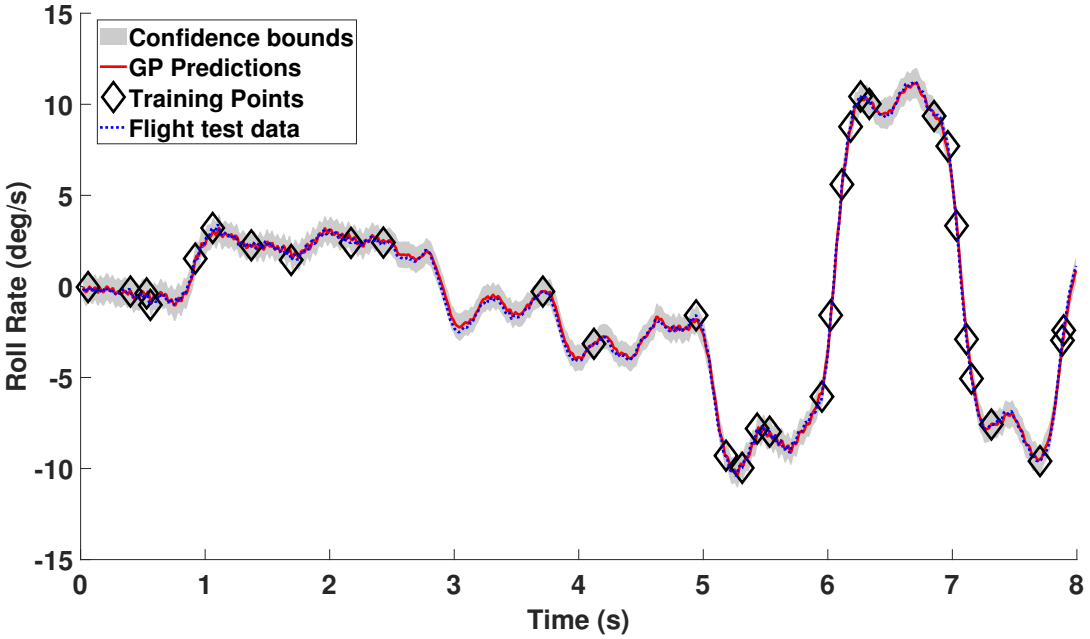


Figure H.2: Sparse Gaussian process one step ahead roll rate prediction using 32 training points and one hyperparameter (α) (Case 2 in Table 8.1).

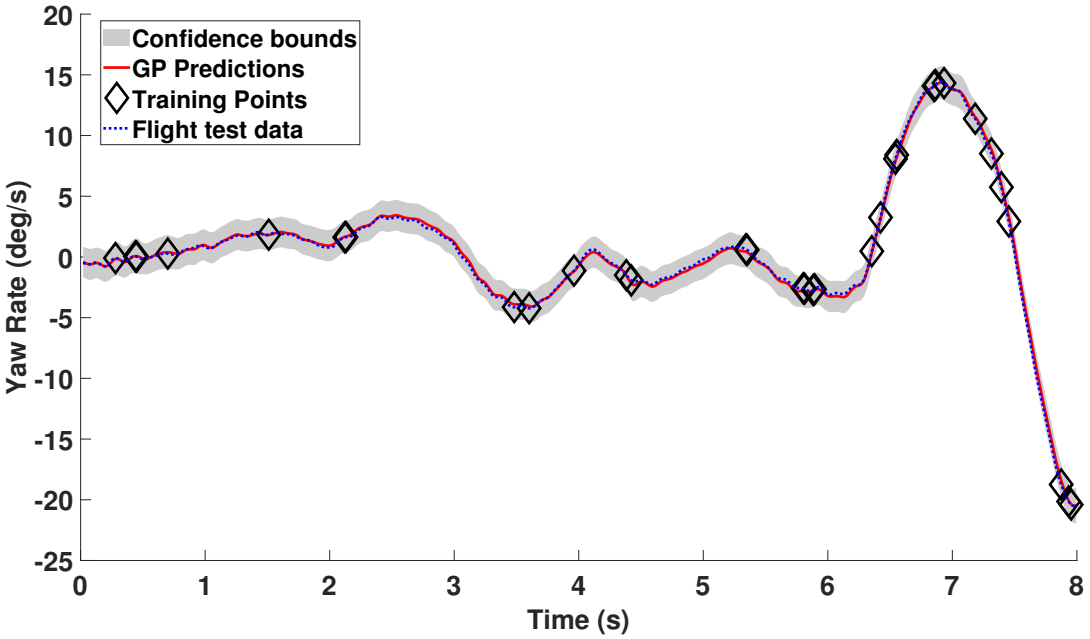


Figure H.3: Sparse Gaussian process one step ahead yaw rate prediction using 32 training points and one hyperparameter (α) (Case 3 in Table 8.1).

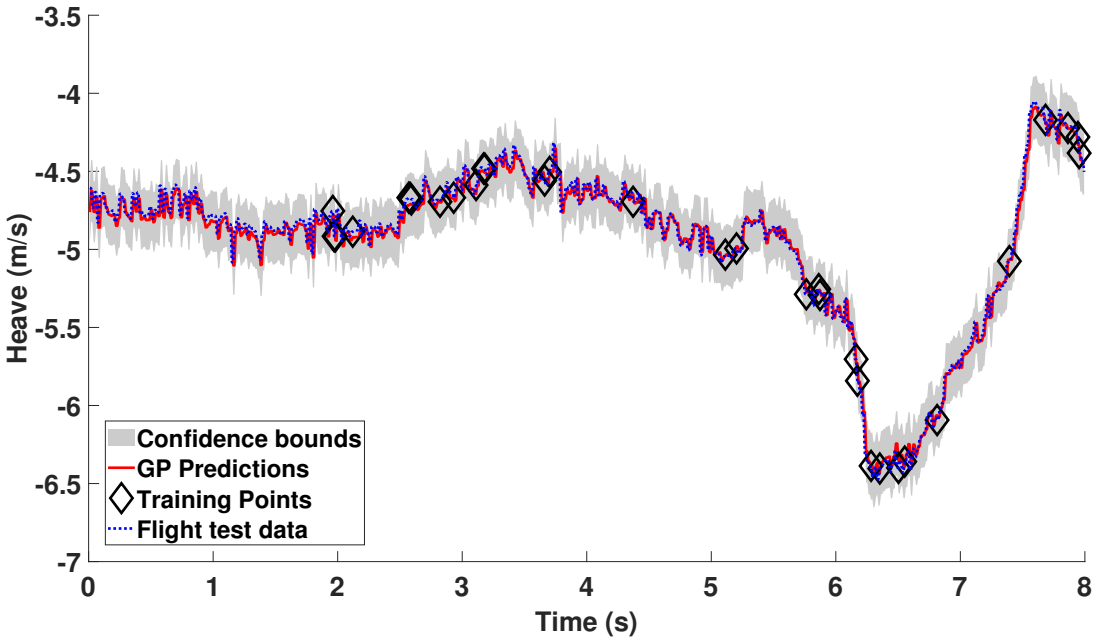


Figure H.4: Sparse Gaussian process one step ahead heave prediction using 32 training points and one hyperparameter (α) (Case 4 in Table 8.1).

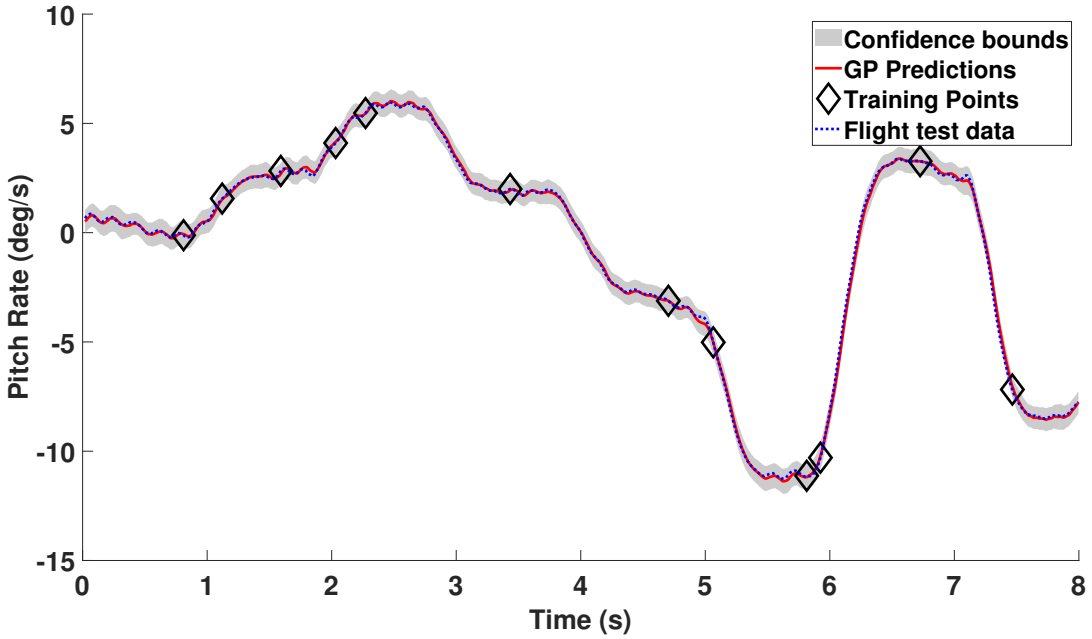


Figure H.5: Sparse Gaussian process one step ahead pitch rate prediction using 12 training points and one hyperparameter (α) (Case 1 in Table 8.1).

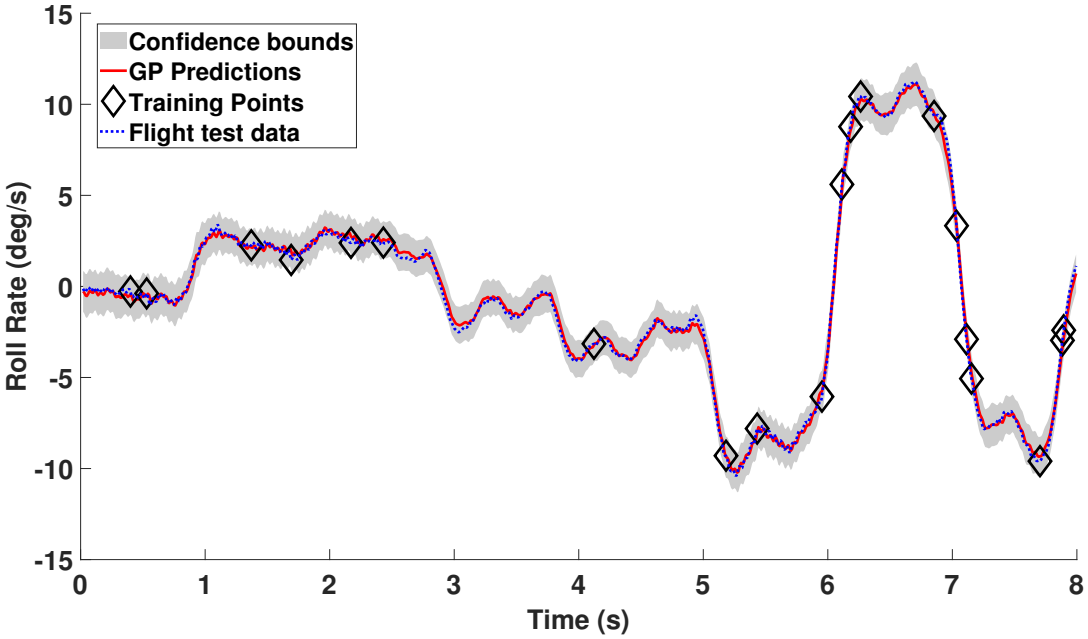


Figure H.6: Sparse Gaussian process one step ahead roll rate prediction using 20 training points and one hyperparameter (α) (Case 2 in Table 8.1).

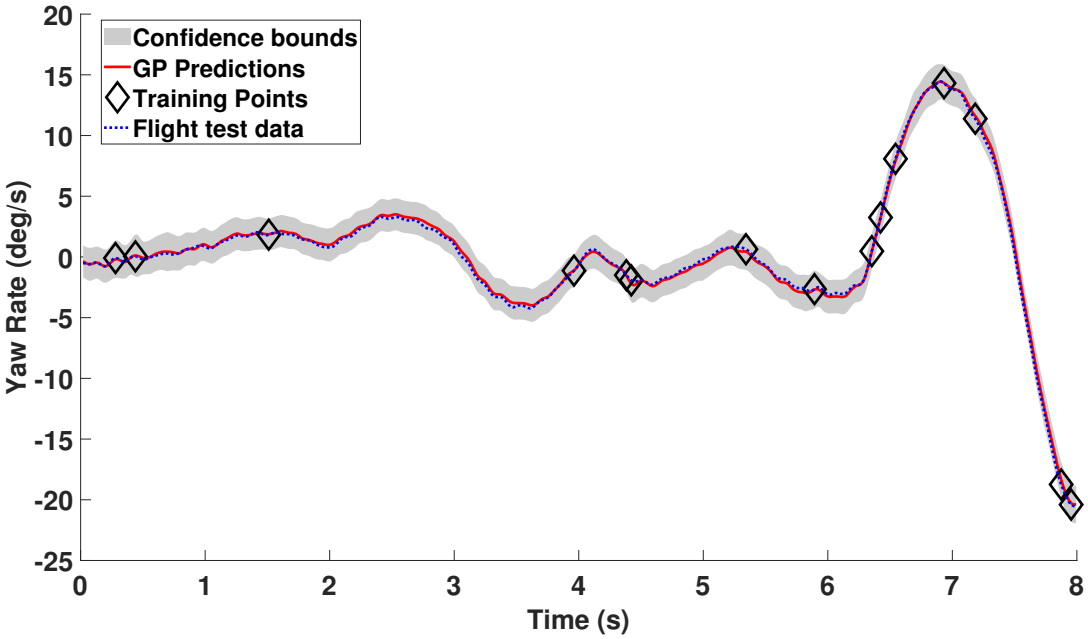


Figure H.7: Sparse Gaussian process one step ahead yaw rate prediction using 15 training points and one hyperparameter (α) (Case 3 in Table 8.1).

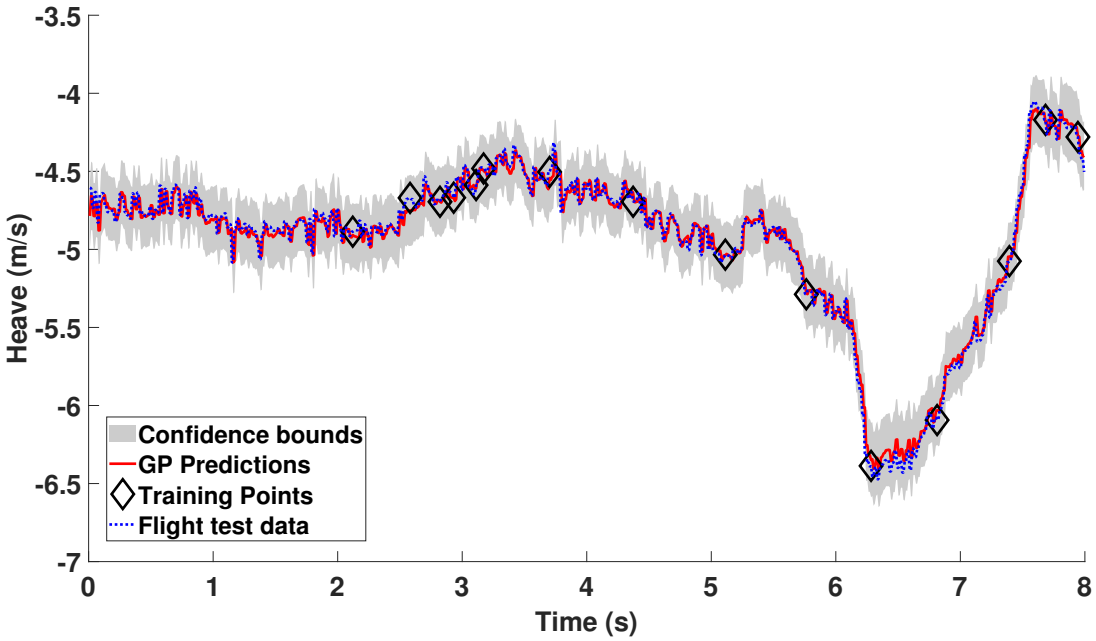


Figure H.8: Sparse Gaussian process one step ahead heave prediction using 15 training points and one hyperparameter (α) (Case 4 in Table 8.1).

Appendix I

Model Error Results

I.1 Linear Regression

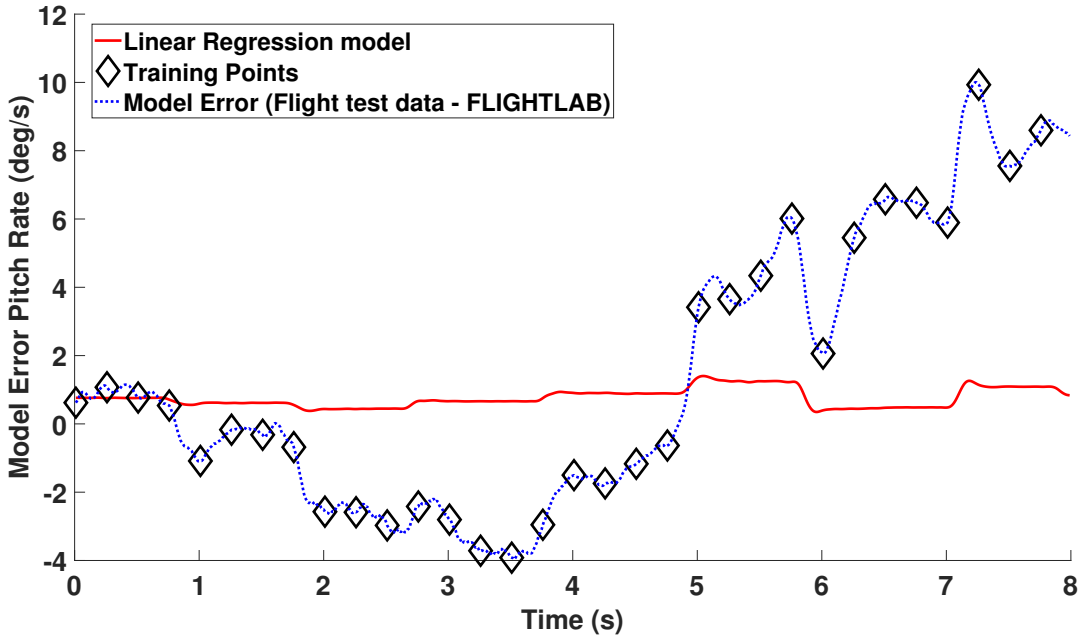


Figure I.1: One step ahead pitch rate predictions with a moving average input structure using a linear regression model.

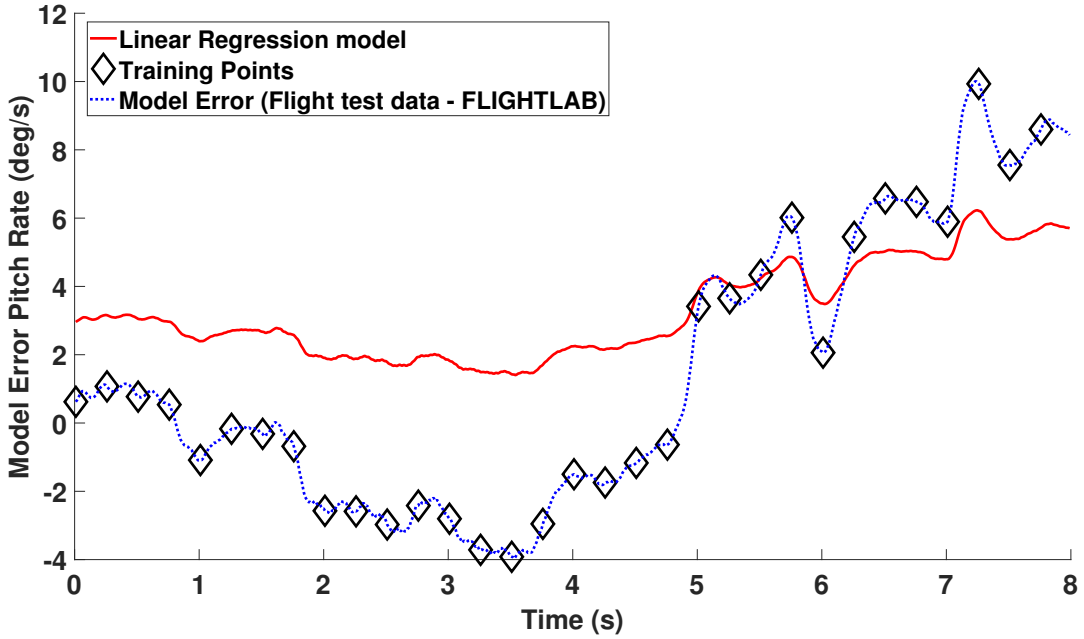


Figure I.2: One step ahead pitch rate predictions with an auto-regressive input structure using a linear regression model.

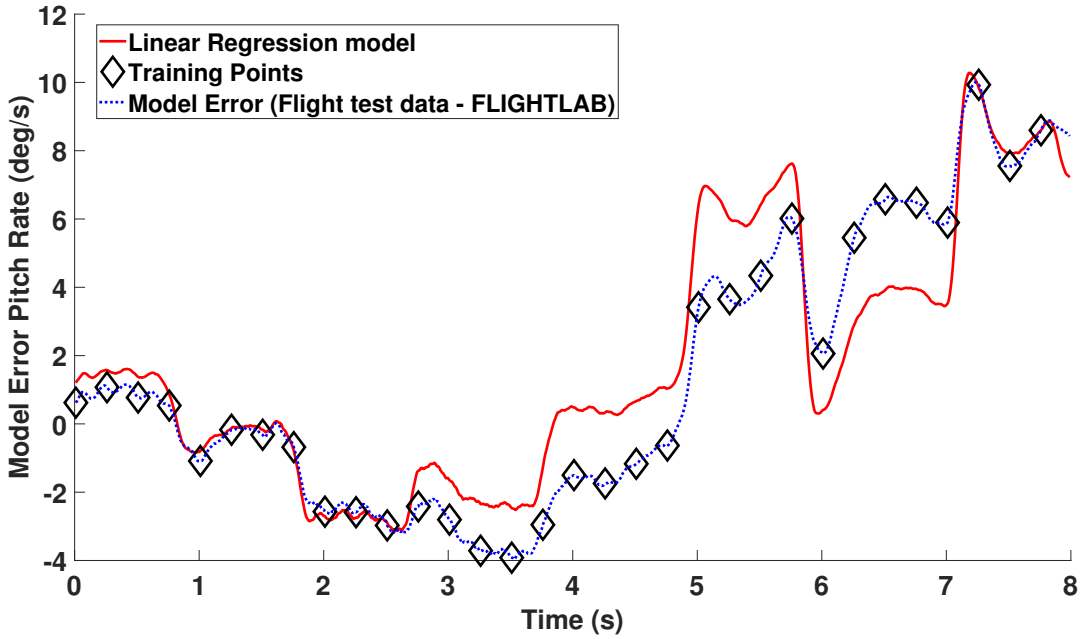


Figure I.3: One step ahead pitch rate predictions with an auto-regressive with exogenous input structure using a linear regression model.

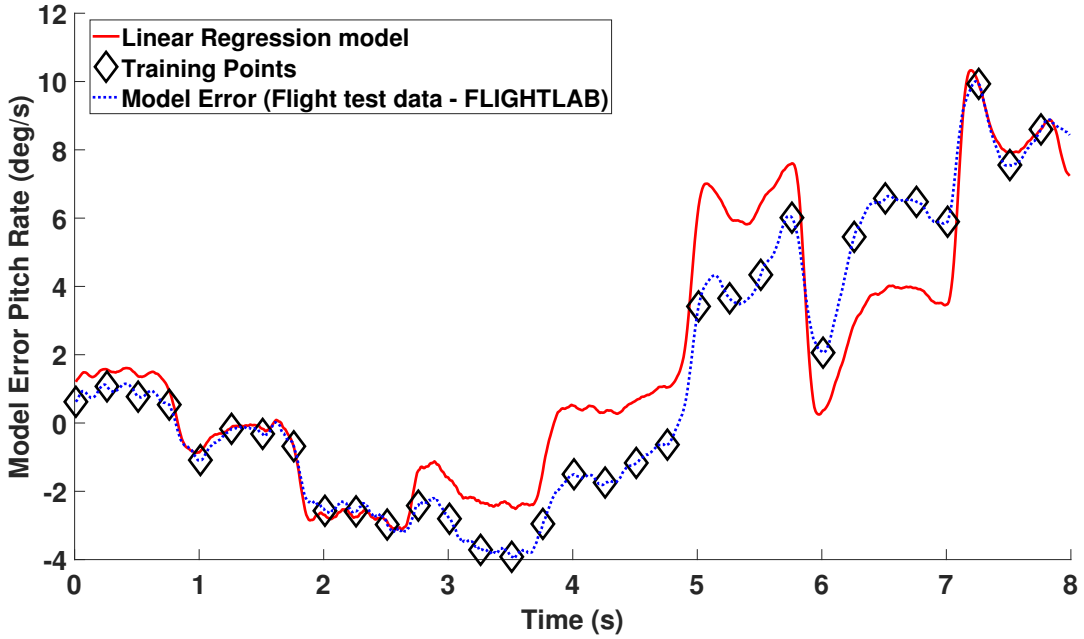


Figure I.4: One step ahead pitch rate predictions with an auto-regressive moving average input structure using a linear regression model.

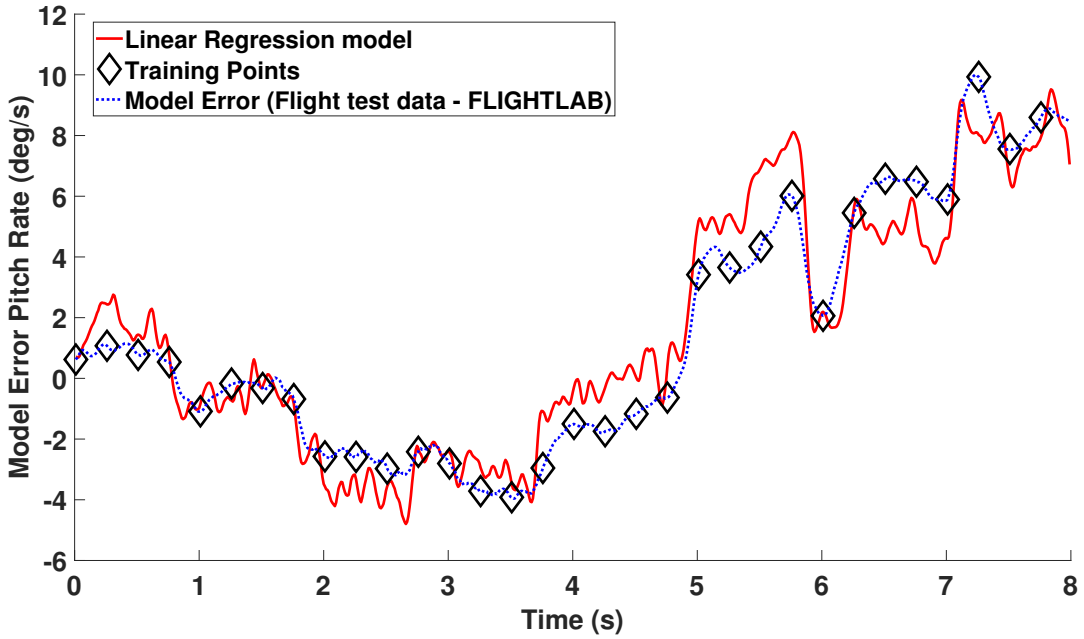


Figure I.5: One step ahead pitch rate predictions with an auto-regressive moving average with exogenous input structure using a linear regression model.

I.2 Investigation of the Gaussian Process input structure

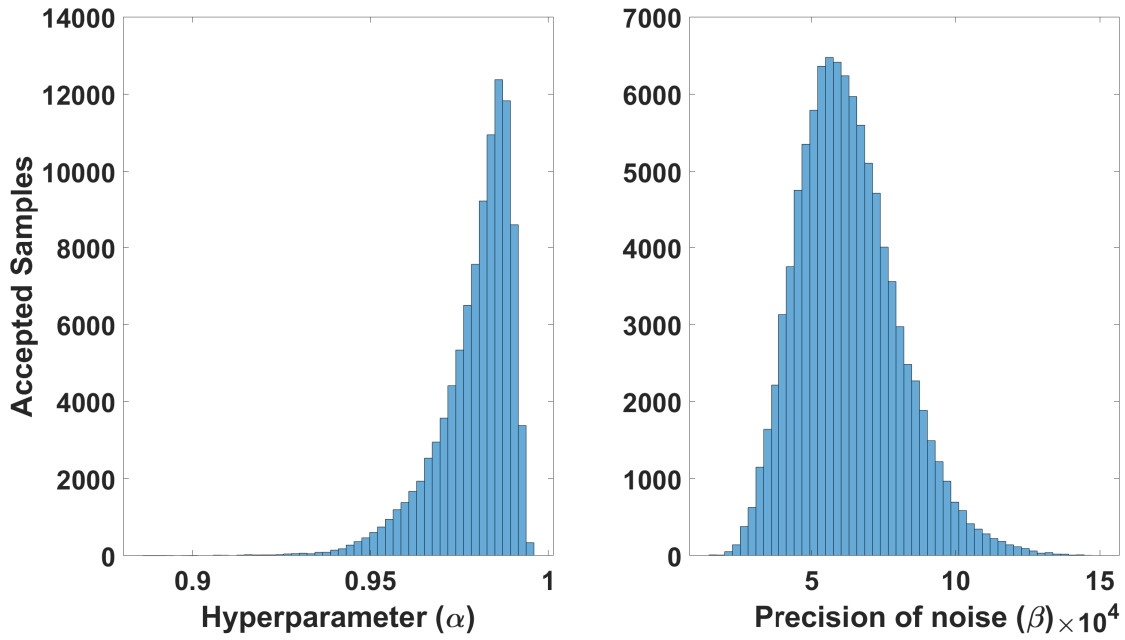


Figure I.6: Simulated annealing results used to create pitch rate model error Gaussian process models, using the auto-regressive with exogenous input structure.

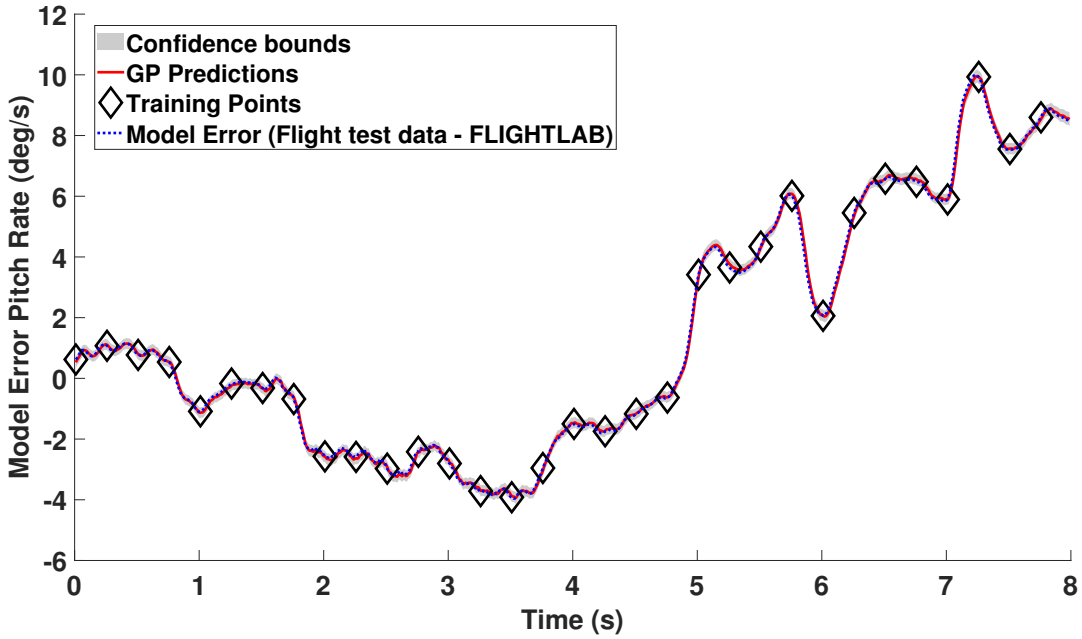


Figure I.7: One step ahead model error pitch rate predictions from a Gaussian process model utilising an auto-regressive with exogenous input structure.

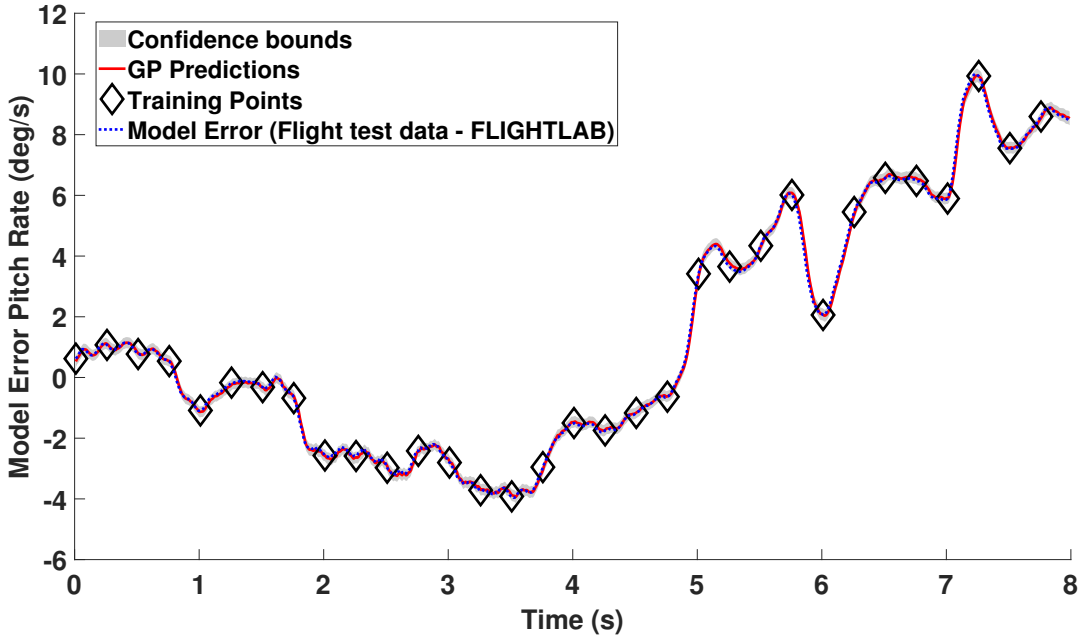


Figure I.8: One step ahead model error pitch rate predictions from a Gaussian process model utilising an auto-regressive moving average input structure.

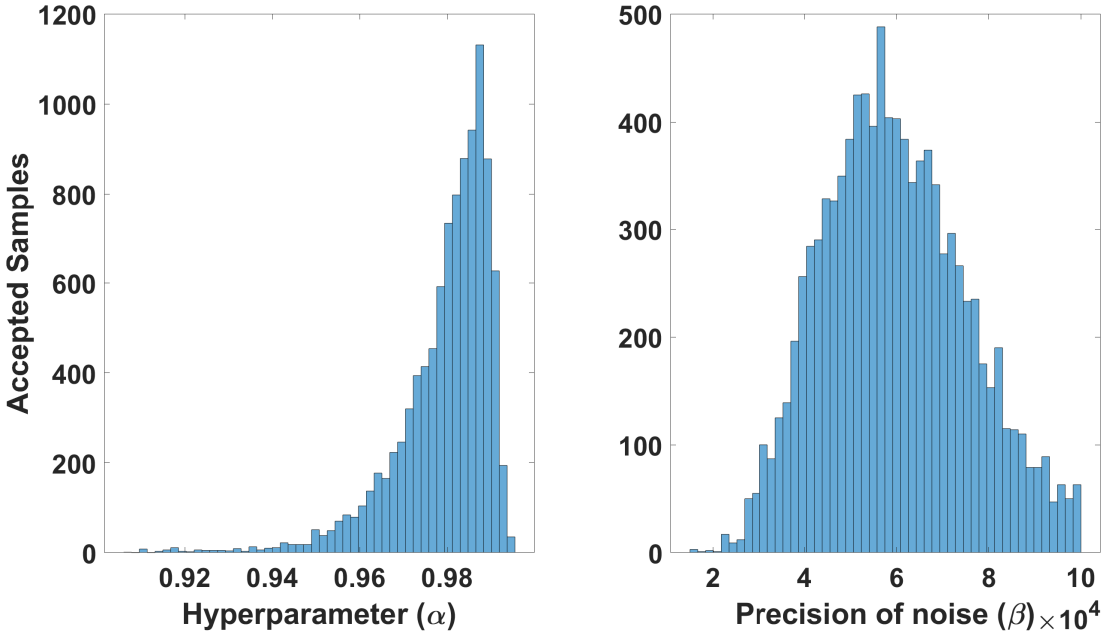


Figure I.9: Simulated annealing results used to create pitch rate model error Gaussian process models, using the auto-regressive moving average input structure.

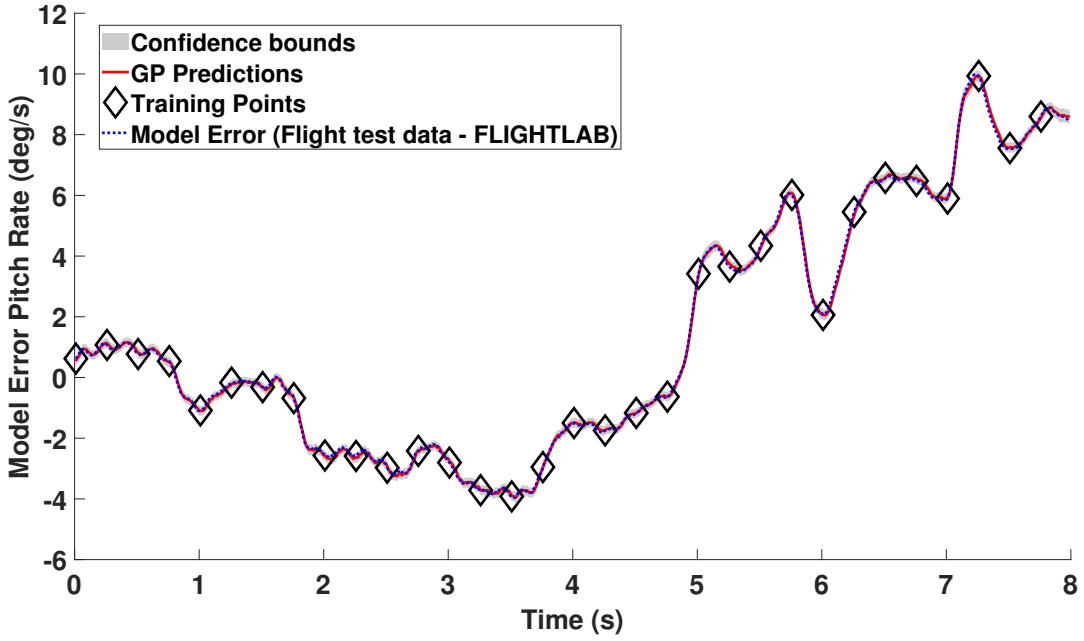


Figure I.10: One step ahead model error pitch rate predictions from a Gaussian process model utilising an auto-regressive moving average with exogenous input structure.

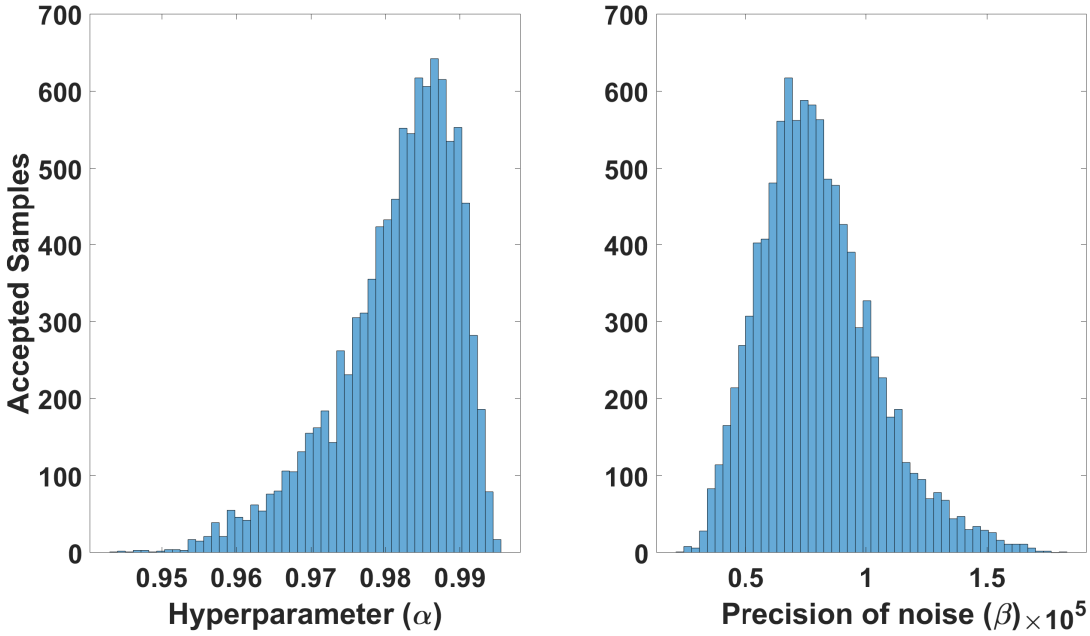


Figure I.11: Simulated annealing results used to create pitch rate model error Gaussian process models, using the auto-regressive moving average with exogenous input structure.

I.3 Pitch Rate

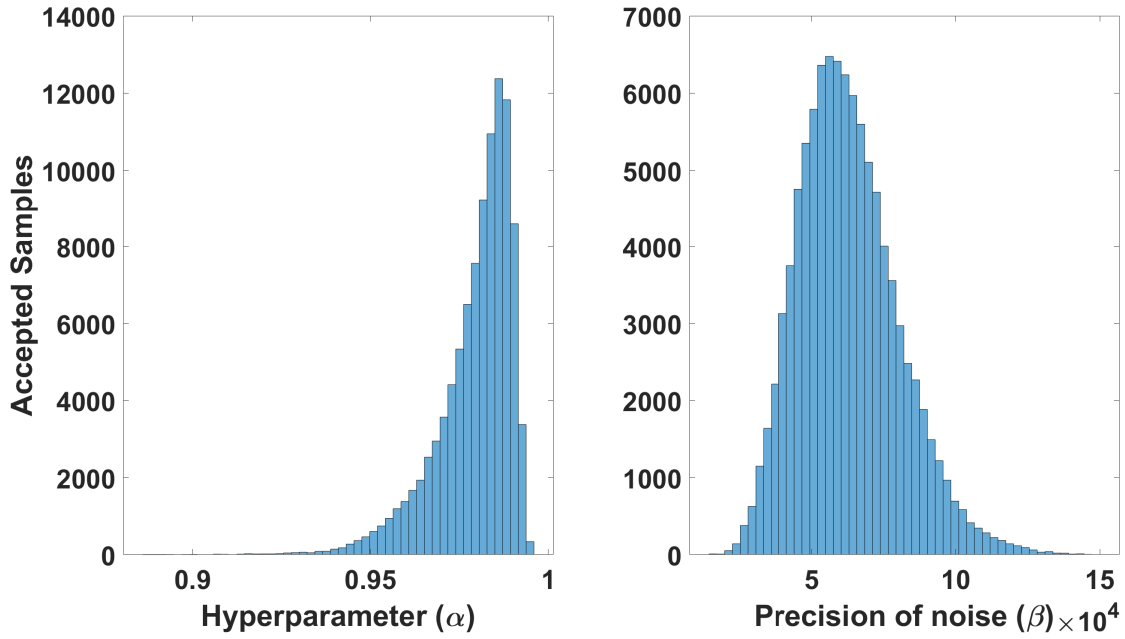


Figure I.12: Simulated annealing results used to create pitch rate model error Gaussian process models, using the input structure containing the current longitudinal pilot input with one lagged model error pitch rate term.

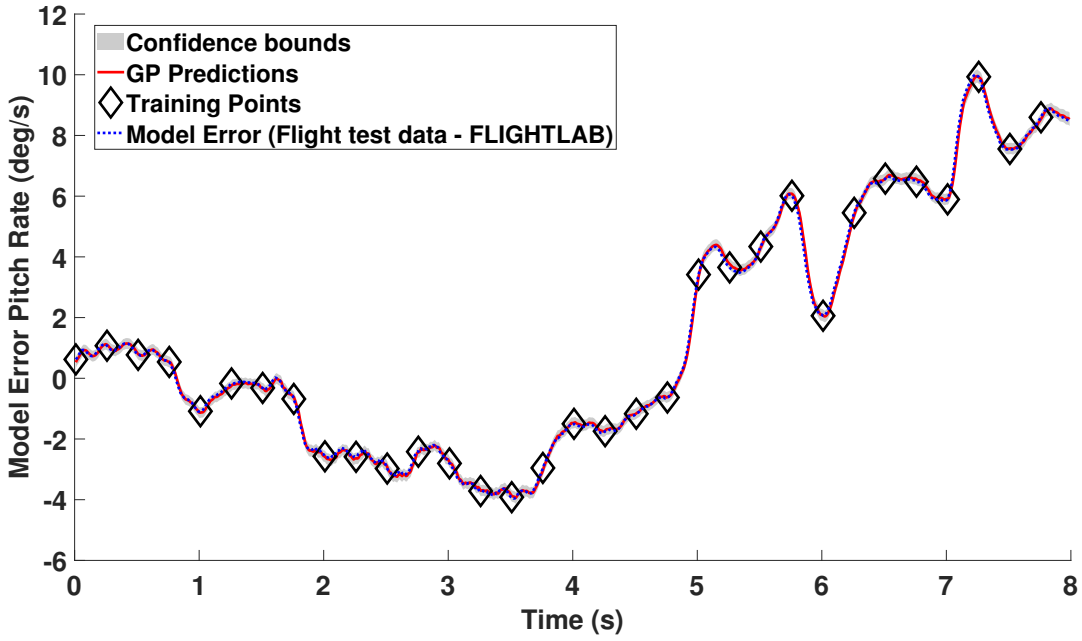


Figure I.13: Gaussian process one step ahead pitch rate model error prediction using the data shown in Figure 9.5, and input structure containing the current longitudinal pilot input with one lagged model error pitch rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.12.

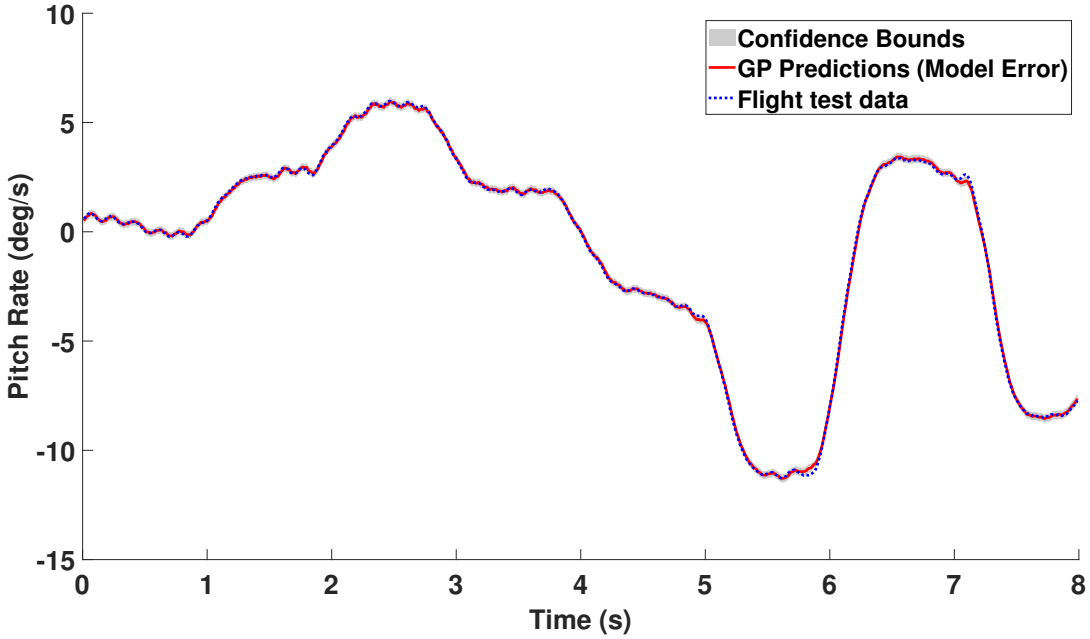


Figure I.14: Gaussian process one step ahead pitch rate prediction using the model error prediction shown in Figure I.13.

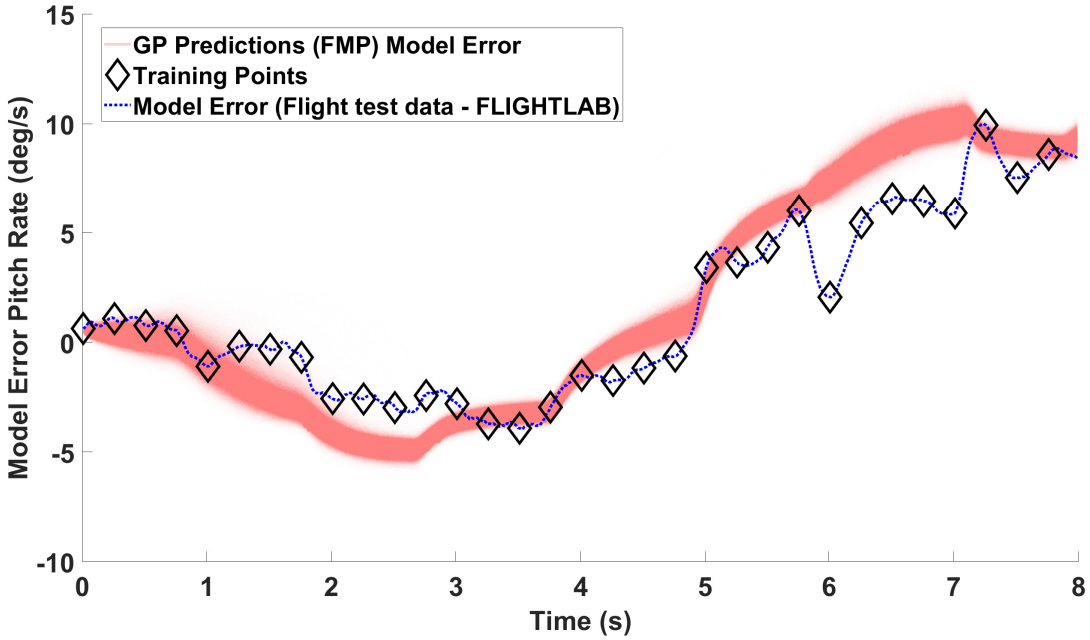


Figure I.15: Realisations of the Gaussian process pitch rate model error full model prediction using the data shown in Figure 9.5, and input structure containing the current longitudinal pilot input with one lagged model error pitch rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.12.

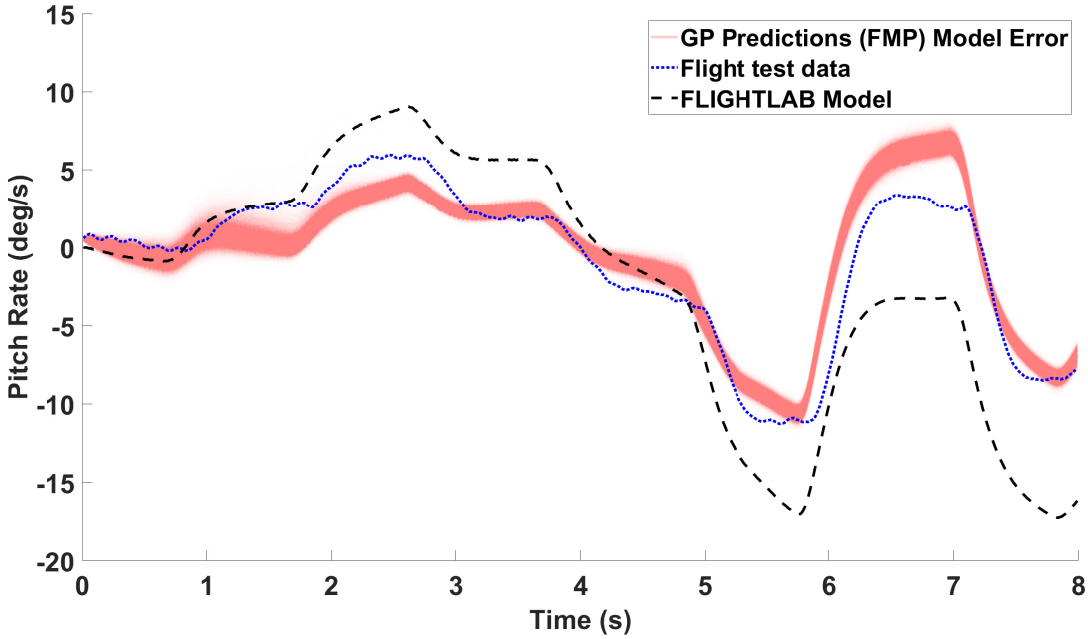


Figure I.16: Realisations of the Gaussian process pitch rate prediction using the model error predictions shown in Figure I.15 with a comparison to the corresponding FLIGHTLAB model.

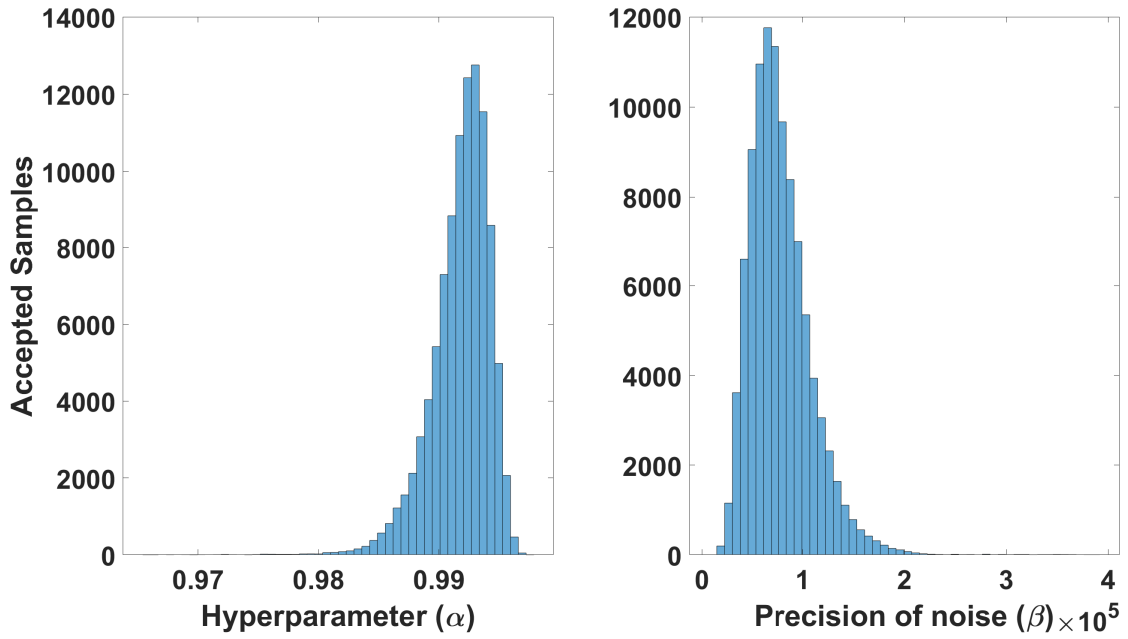


Figure I.17: Simulated annealing results used to create pitch rate model error Gaussian process models, using the input structure containing all of the current pilot inputs with one lagged model error pitch rate term.

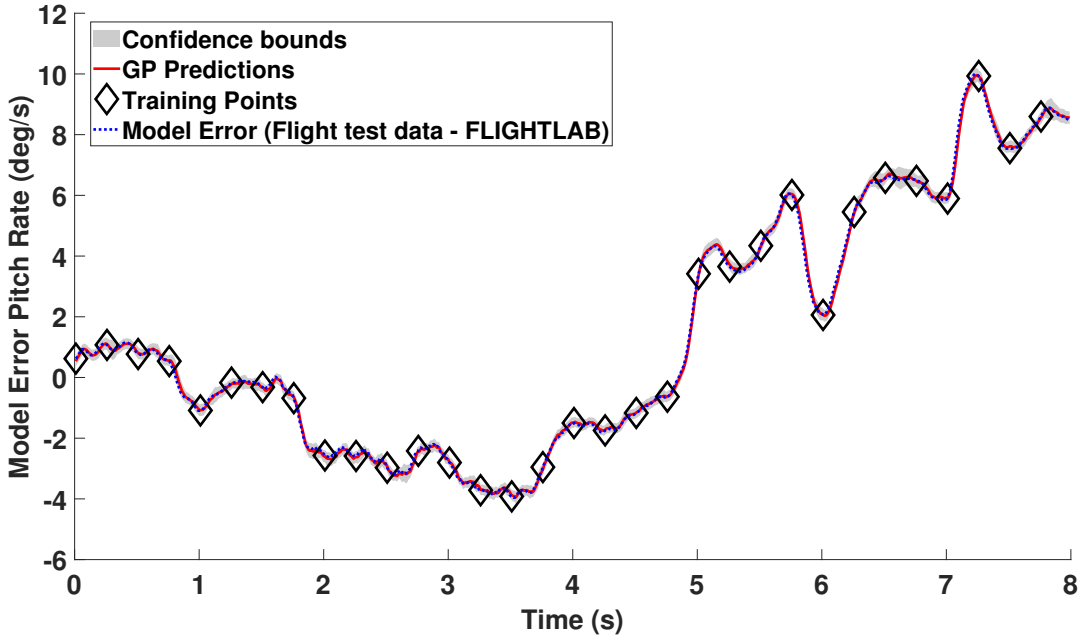


Figure I.18: Gaussian process one step ahead pitch rate model error prediction using the data shown in Figure 9.5 and input structure containing all of the current pilot inputs with one lagged model error pitch rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.17.

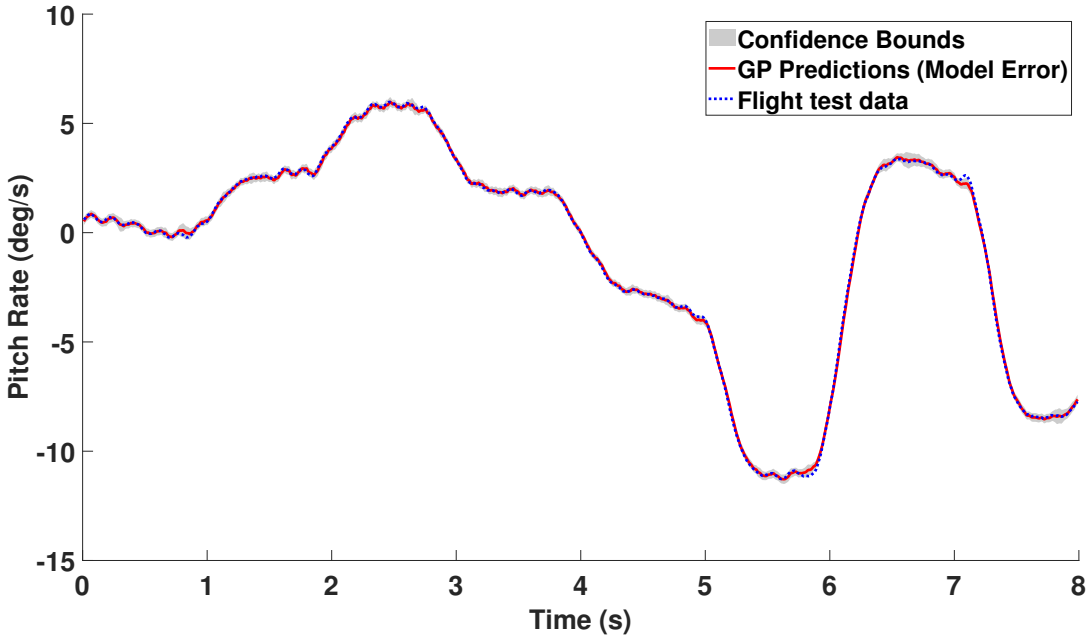


Figure I.19: Gaussian process one step ahead pitch rate prediction using the model error prediction shown in Figure I.18.

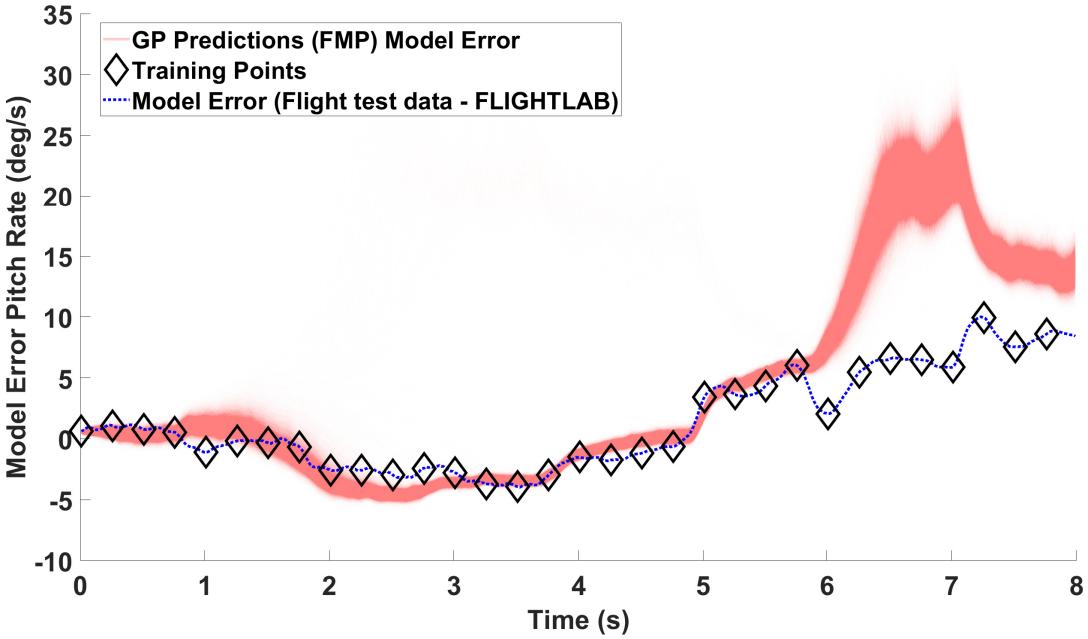


Figure I.20: Realisations of the Gaussian process pitch rate model error full model prediction using the data shown in Figure 9.5 and input structure containing all of the current pilot inputs with one lagged model error pitch rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.17.

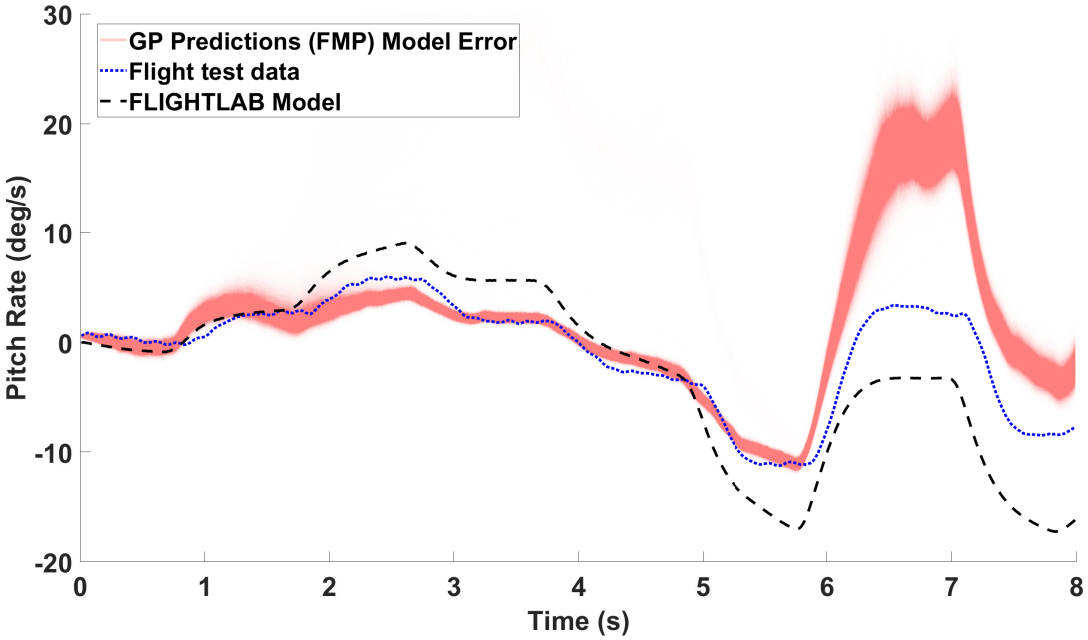


Figure I.21: Realisations of the pitch rate prediction using the model error predictions shown in Figure I.20 with a comparison to the corresponding FLIGHTLAB model.

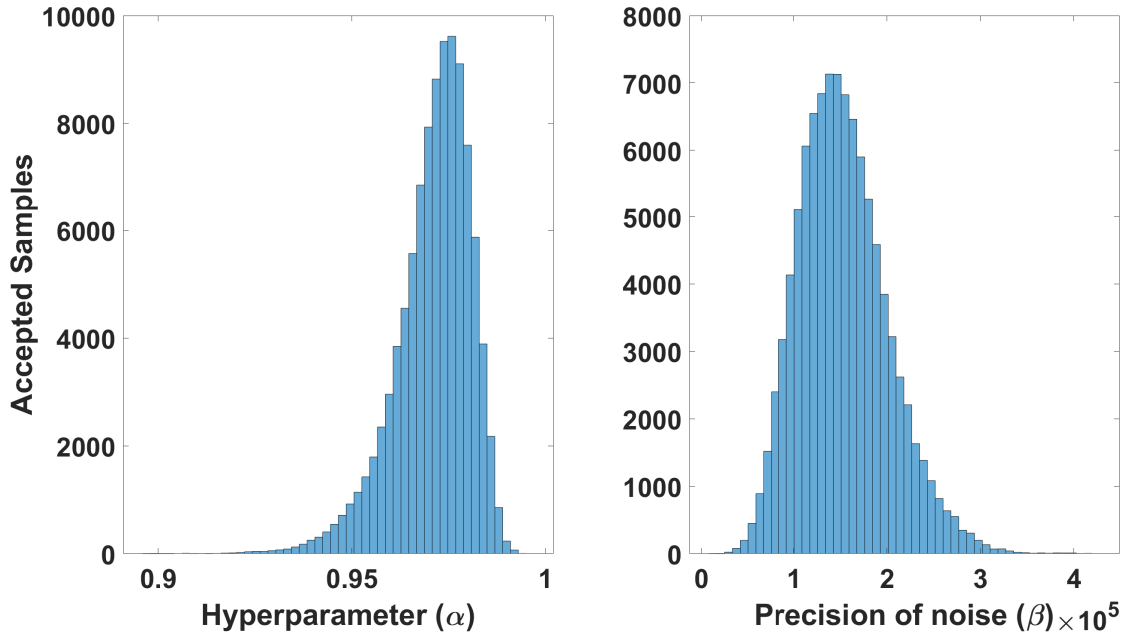


Figure I.22: Simulated annealing results used to create pitch rate model error Gaussian process models, using the input structure containing the current longitudinal pilot input with two lagged model error pitch rate terms.

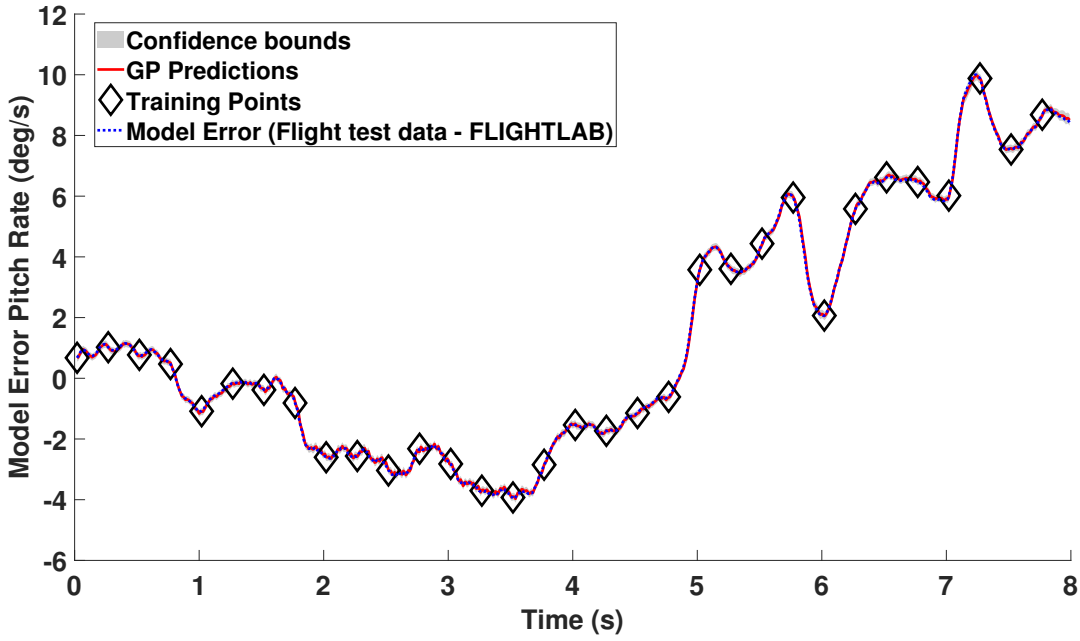


Figure I.23: Gaussian process one step ahead pitch rate model error prediction using the data shown in Figure 9.5 and input structure containing the current longitudinal pilot input with two lagged model error pitch rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.22.

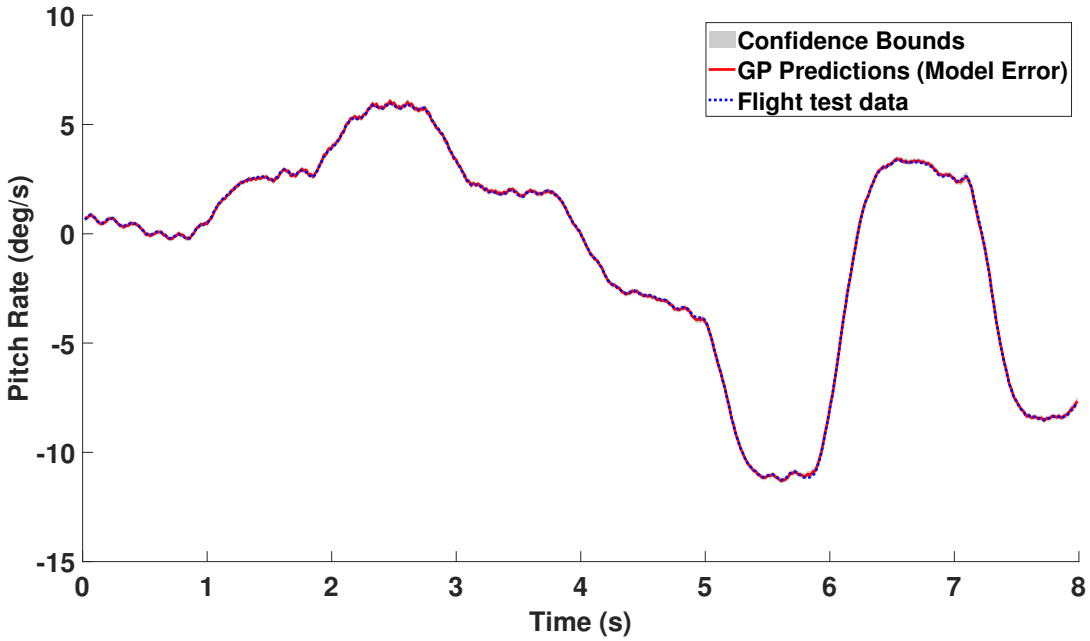


Figure I.24: Gaussian process one step ahead pitch rate prediction using the model error prediction shown in Figure I.23.

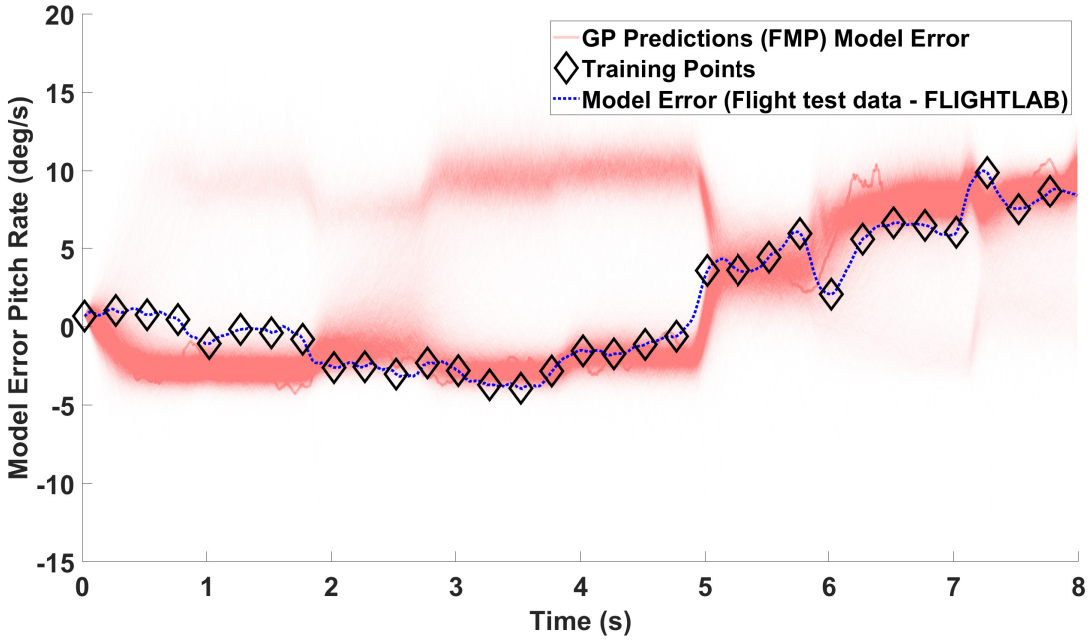


Figure I.25: Realisations of the Gaussian process pitch rate model error full model prediction using the data shown in Figure 9.5 and input structure containing the current longitudinal pilot input with two lagged model error pitch rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.22.

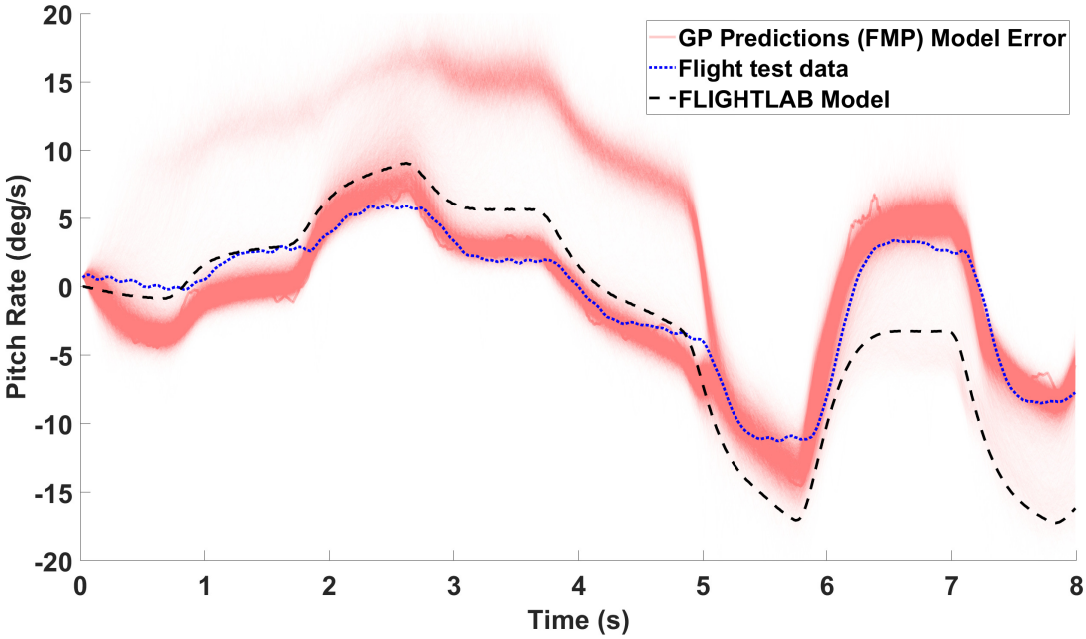


Figure I.26: Realisations of the Gaussian process pitch rate prediction using the model error predictions shown in Figure I.25 with a comparison to the corresponding FLIGHTLAB model.

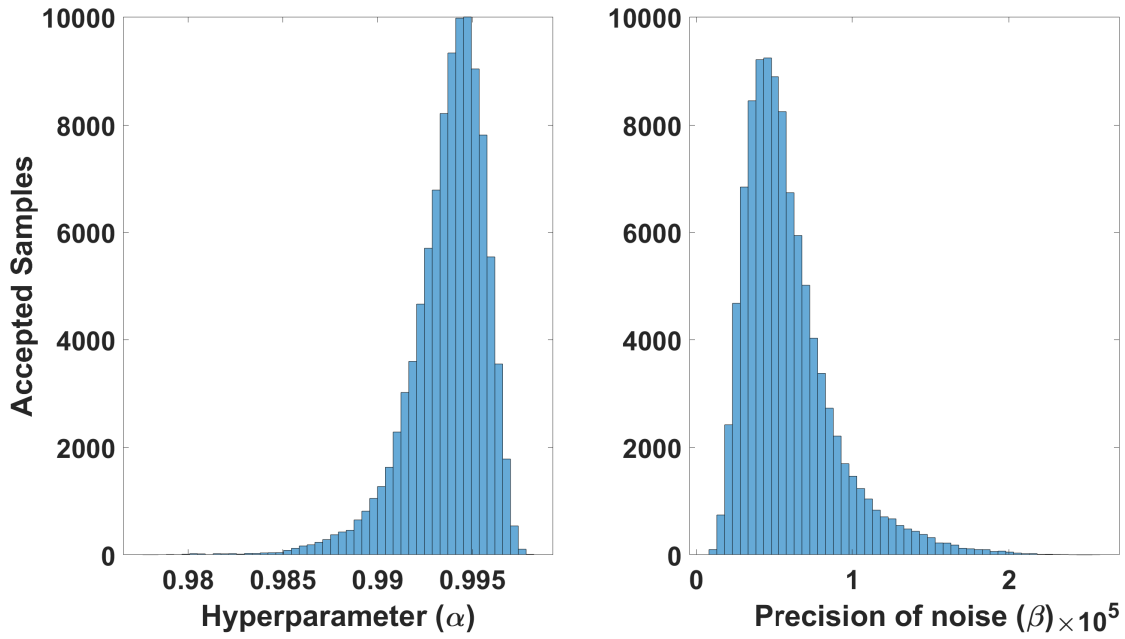


Figure I.27: Simulated annealing results used to create pitch rate model error Gaussian process models, using the input structure containing all of the current pilot inputs with two lagged model error pitch rate terms.

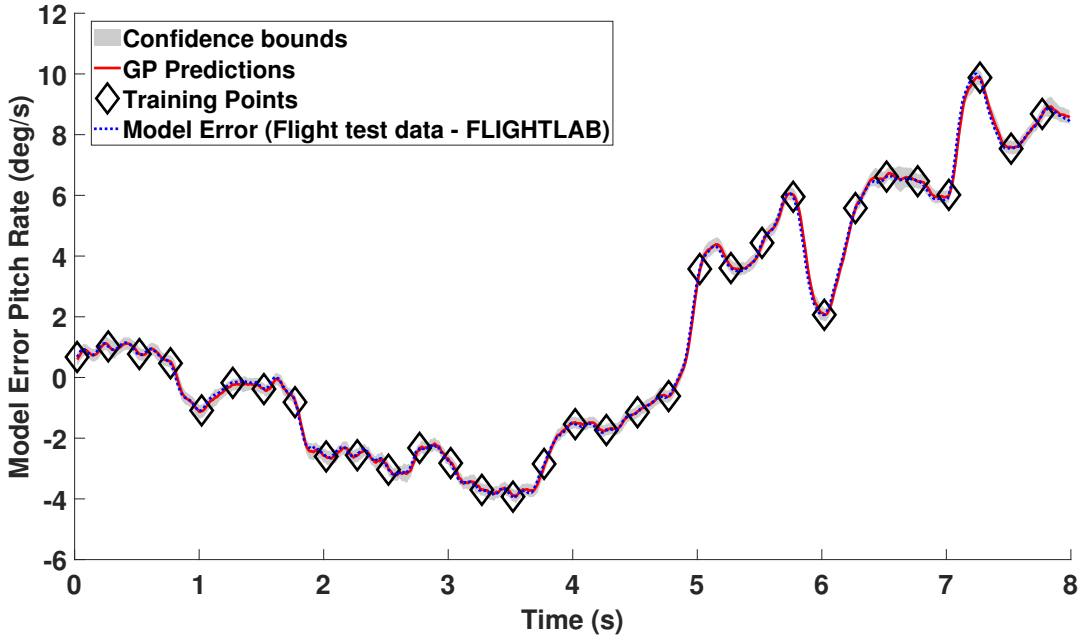


Figure I.28: Gaussian process one step ahead pitch rate model error prediction using the data shown in Figure 9.5 and input structure containing all of the current pilot inputs with two lagged model error pitch rate terms and one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.27.

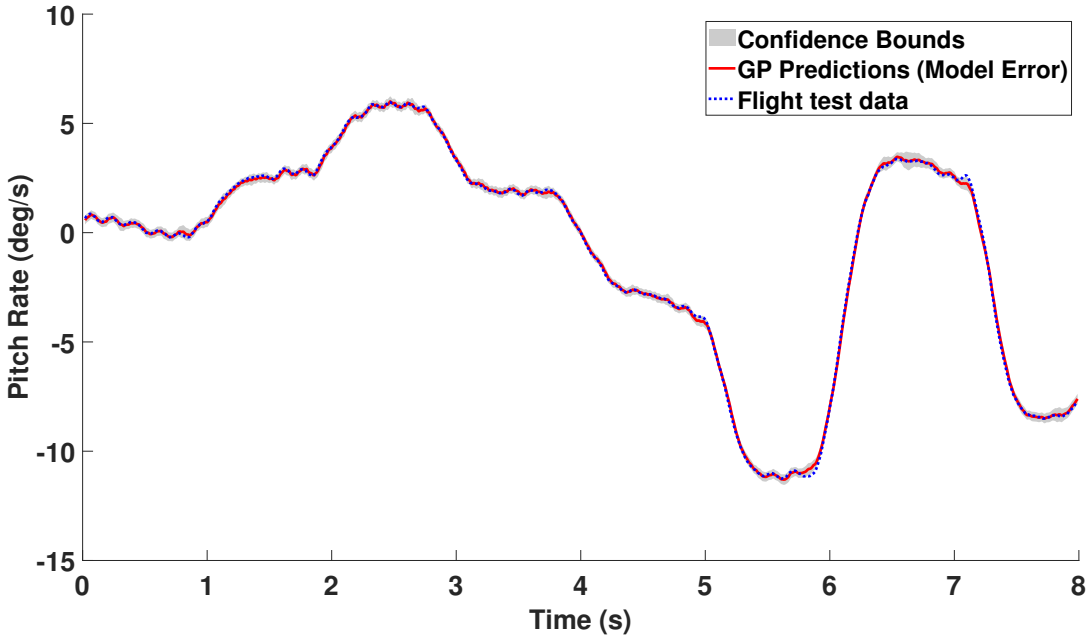


Figure I.29: Gaussian process one step ahead pitch rate prediction using the model error prediction shown in Figure I.28.

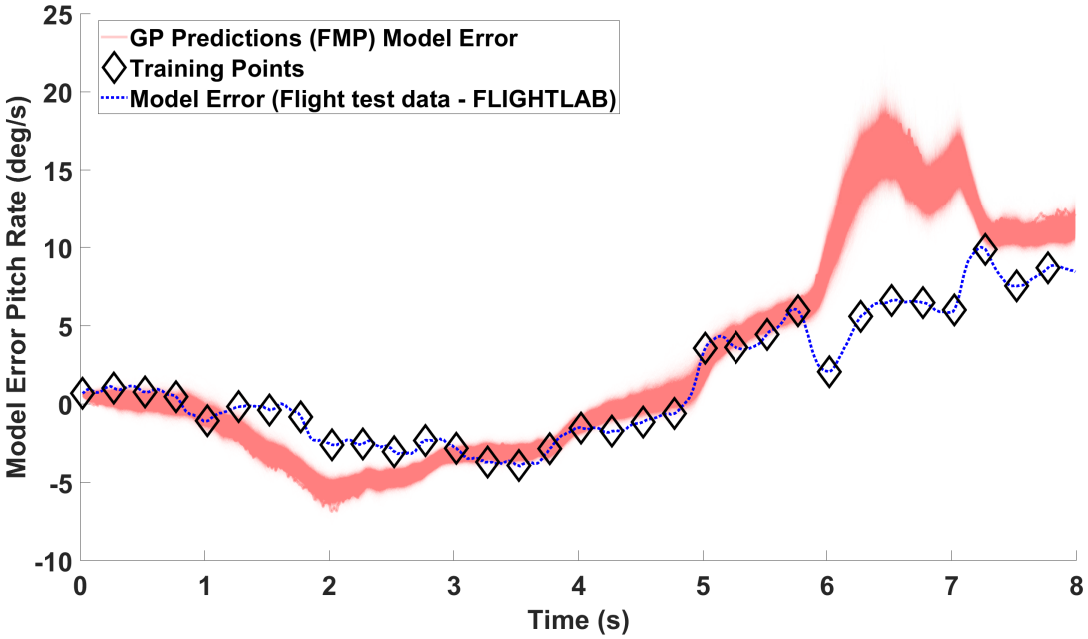


Figure I.30: Realisations of the Gaussian process pitch rate model error full model prediction using the data shown in Figure 9.5 and input structure containing all of the current pilot inputs with two lagged model error pitch rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.27.

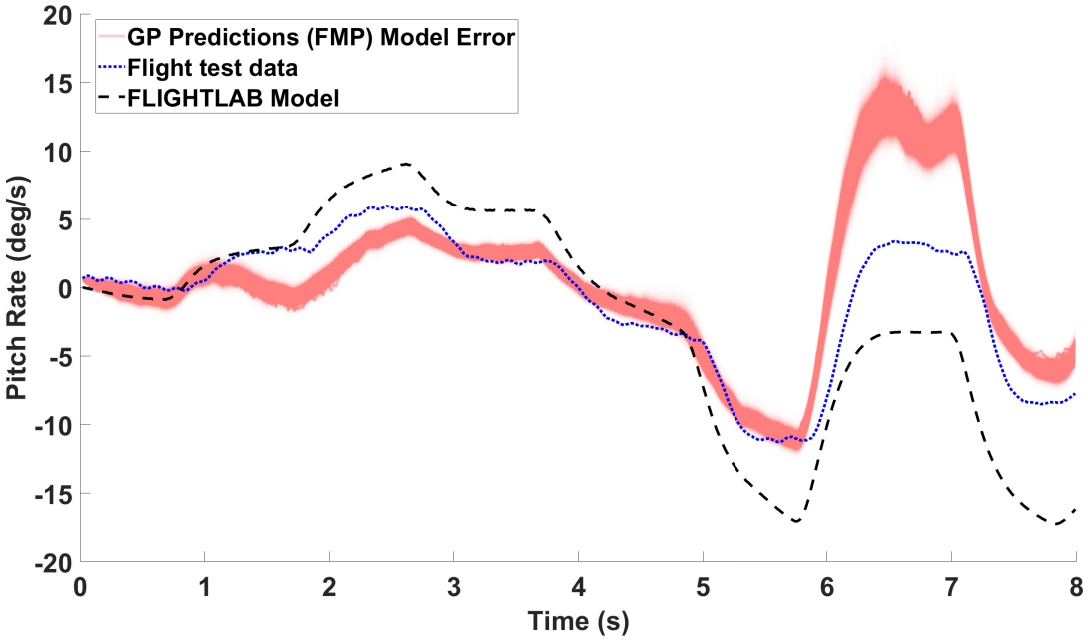


Figure I.31: Realisations of the pitch rate prediction using the model error predictions shown in Figure I.30 with a comparison to the corresponding FLIGHTLAB model.

I.4 Roll Rate

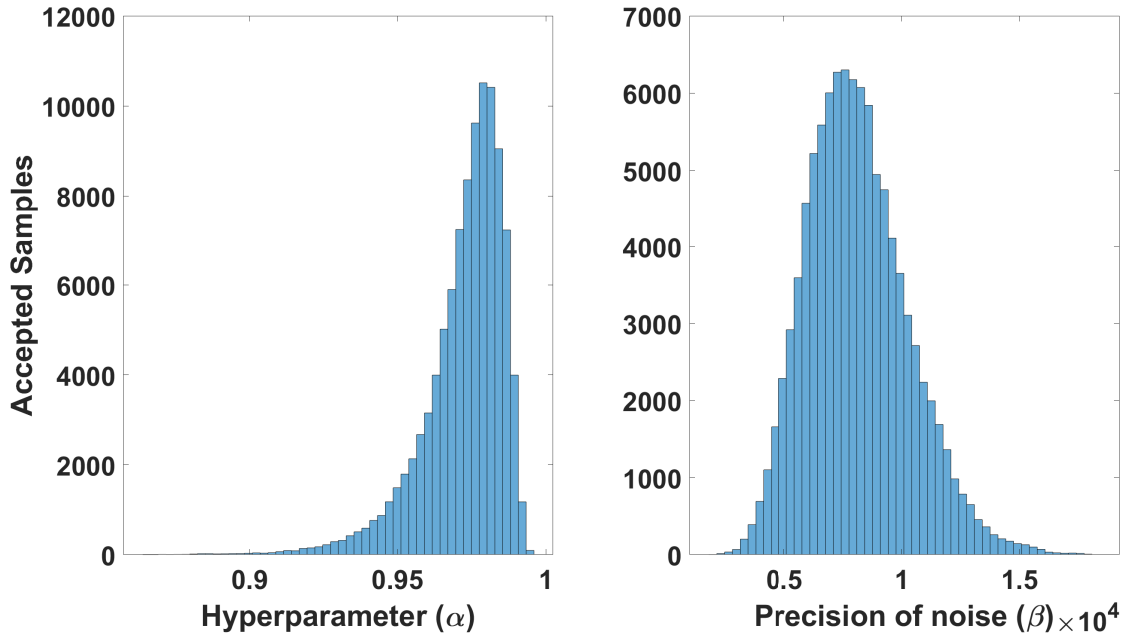


Figure I.32: Simulated annealing results used to create roll rate model error Gaussian process models, using the input structure containing the current lateral pilot input with one lagged model error roll rate term.

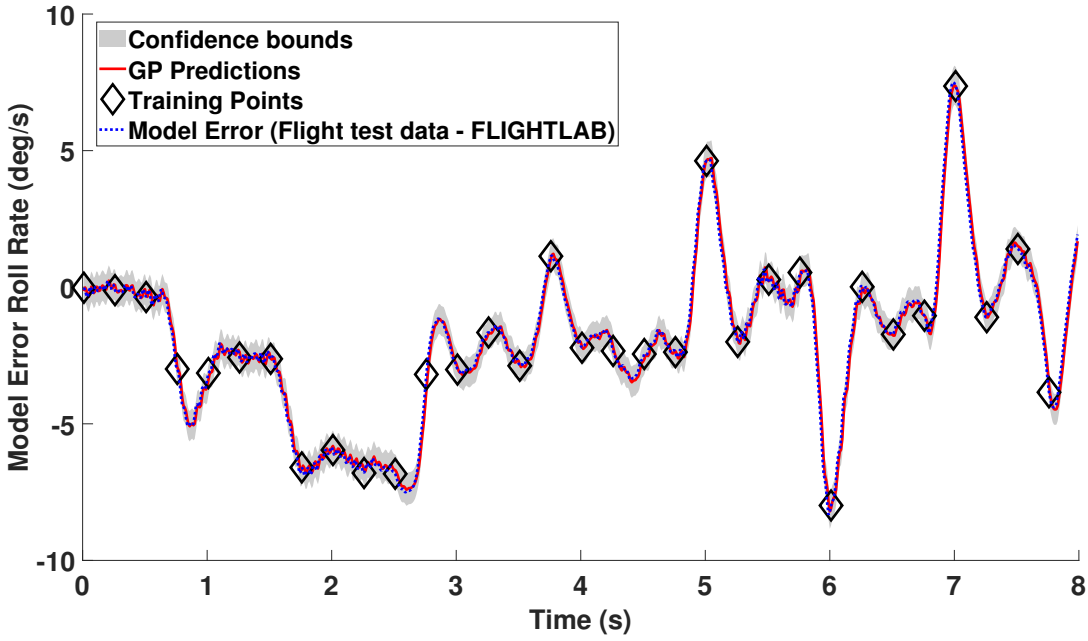


Figure I.33: Gaussian process one step ahead roll rate model error prediction using the data shown in Figure 9.9 and input structure containing the current lateral pilot input with one lagged model error roll rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.32.

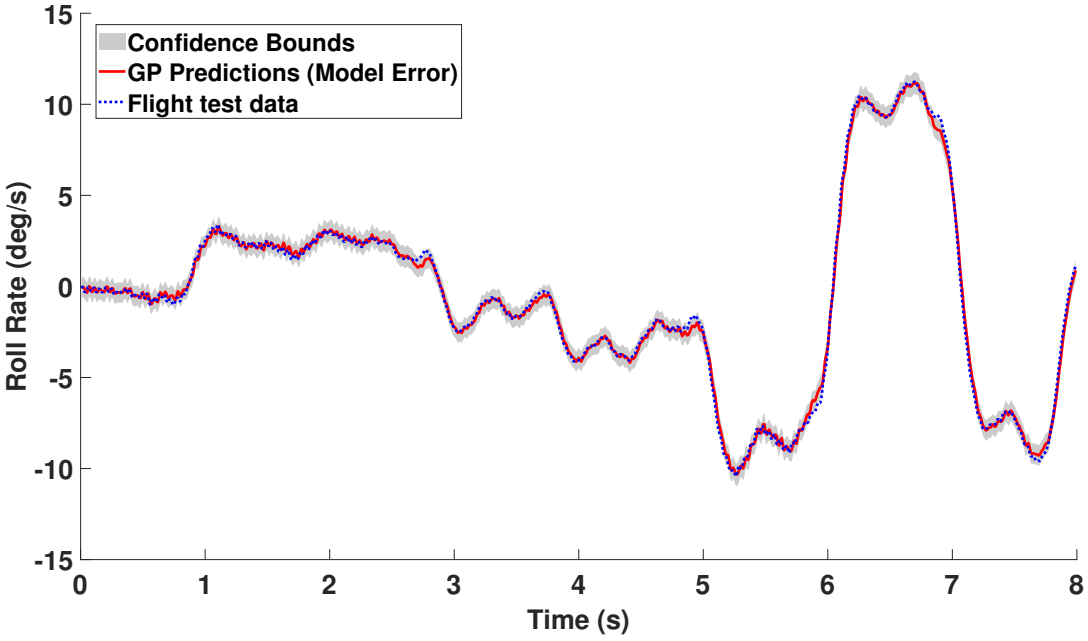


Figure I.34: Gaussian process one step ahead roll rate prediction using the model error prediction shown in Figure I.33.

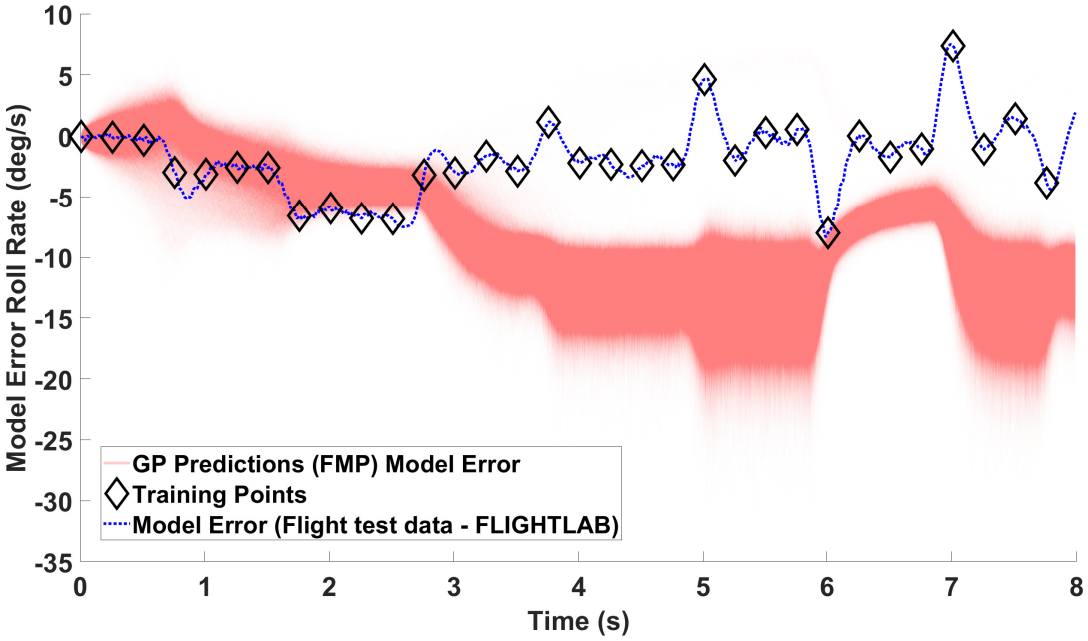


Figure I.35: Realisations of the Gaussian process roll rate model error full model prediction using the data shown in Figure 9.9 and input structure containing the current lateral pilot input with one lagged model error roll rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.32.

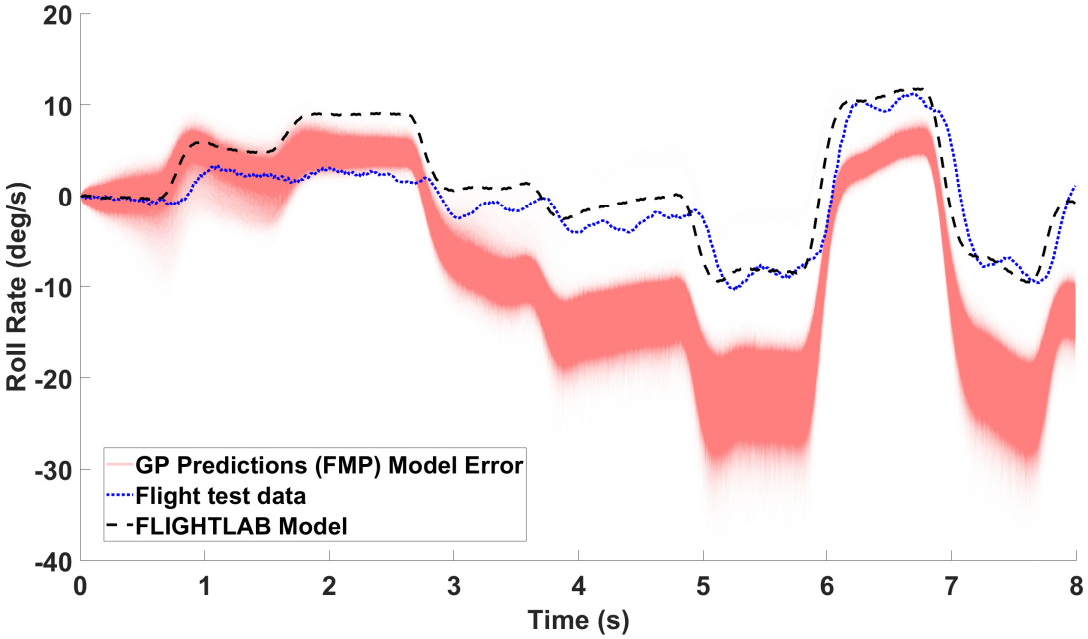


Figure I.36: Realisations of the roll rate prediction using the model error predictions shown in Figure I.35 with a comparison to the corresponding FLIGHTLAB model.

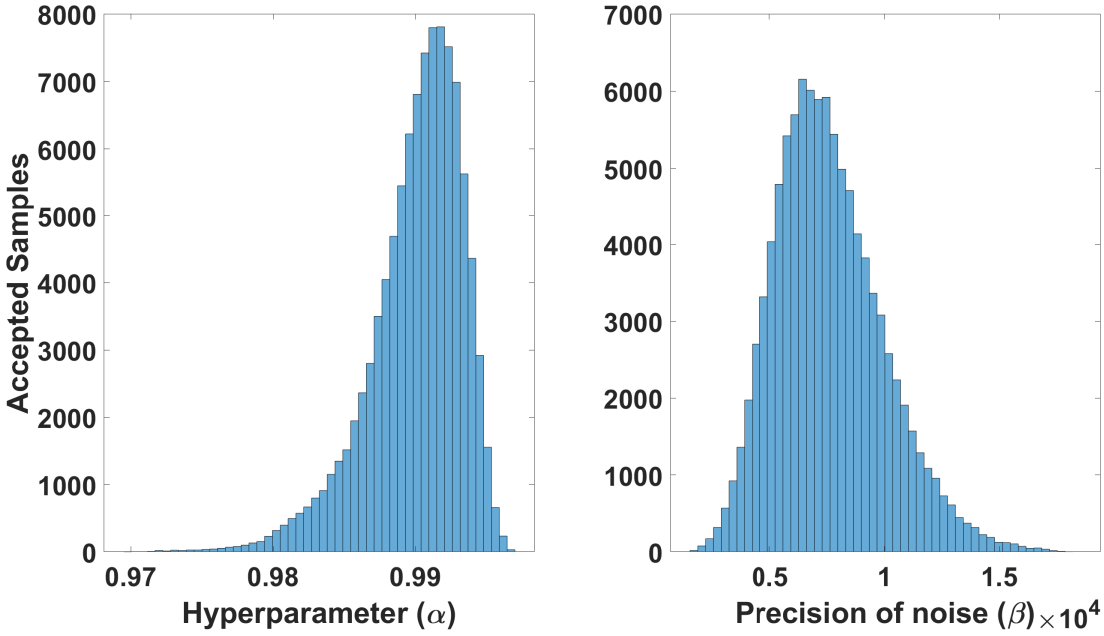


Figure I.37: Simulated annealing results used to create roll rate model error Gaussian process models, using the input structure containing all of the current pilot inputs with one lagged model error roll rate term.

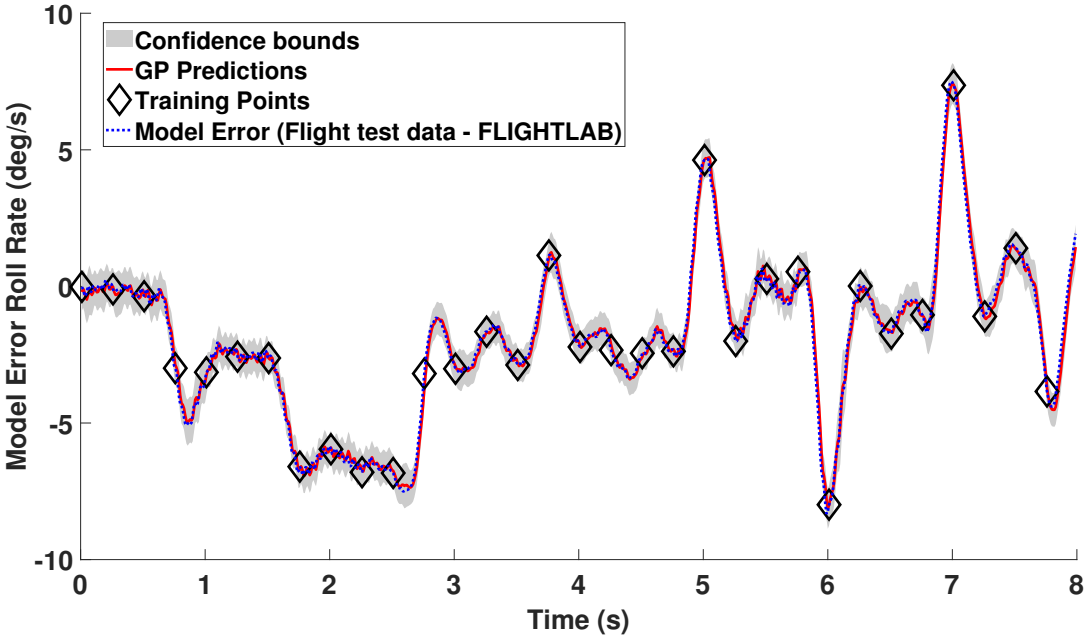


Figure I.38: Gaussian process one step ahead roll rate model error prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with one lagged model error roll rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.37.

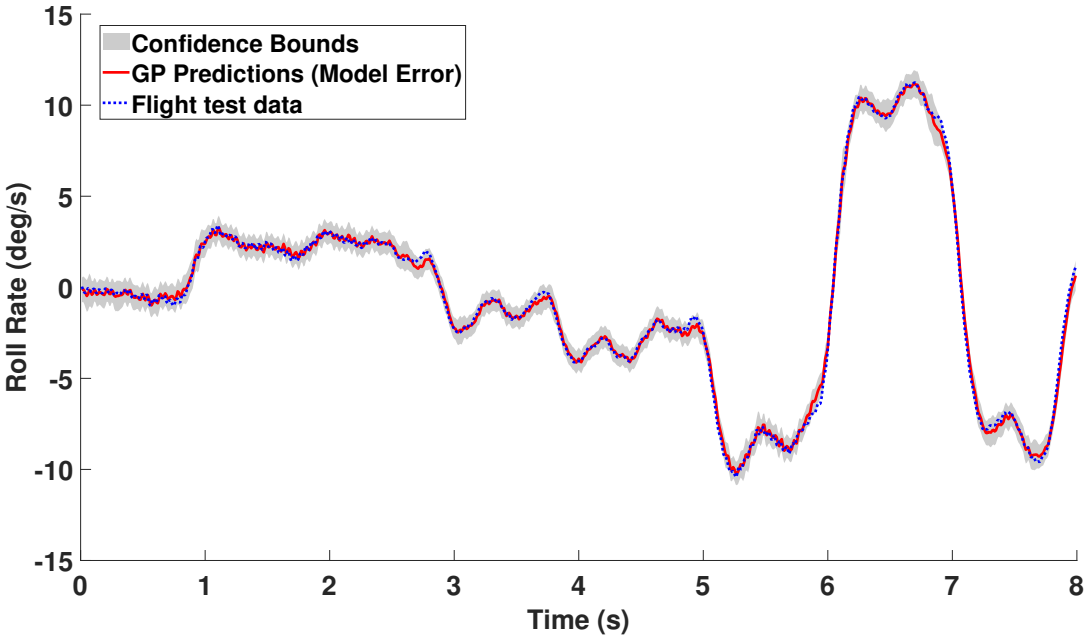


Figure I.39: Gaussian process one step ahead roll rate prediction using the model error prediction shown in Figure I.38.

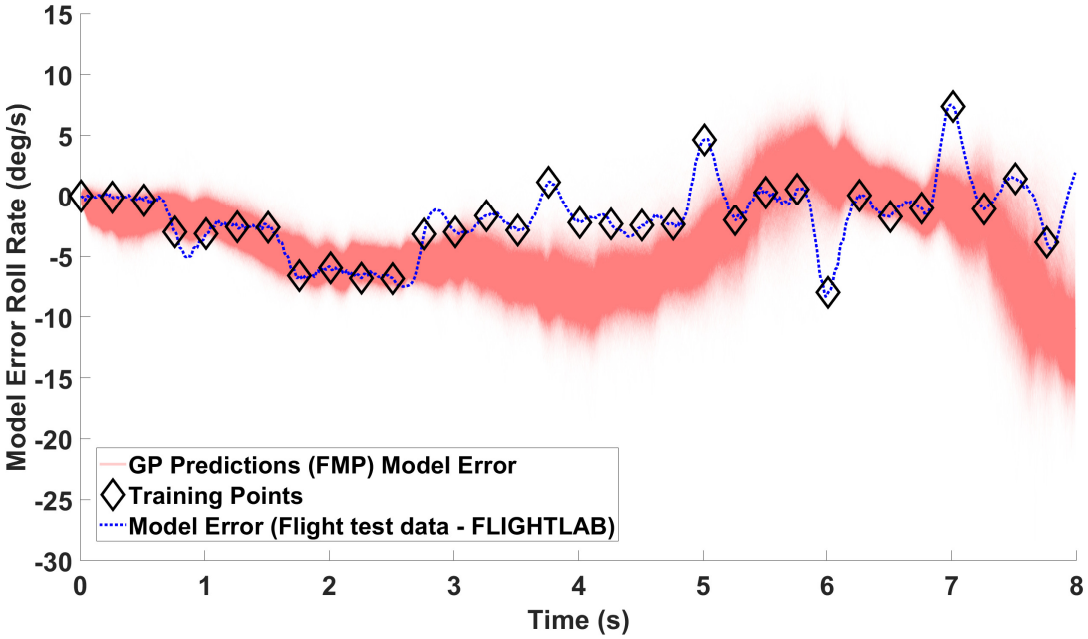


Figure I.40: Realisations of the Gaussian process roll rate model error full model prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with one lagged model error roll rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.37.

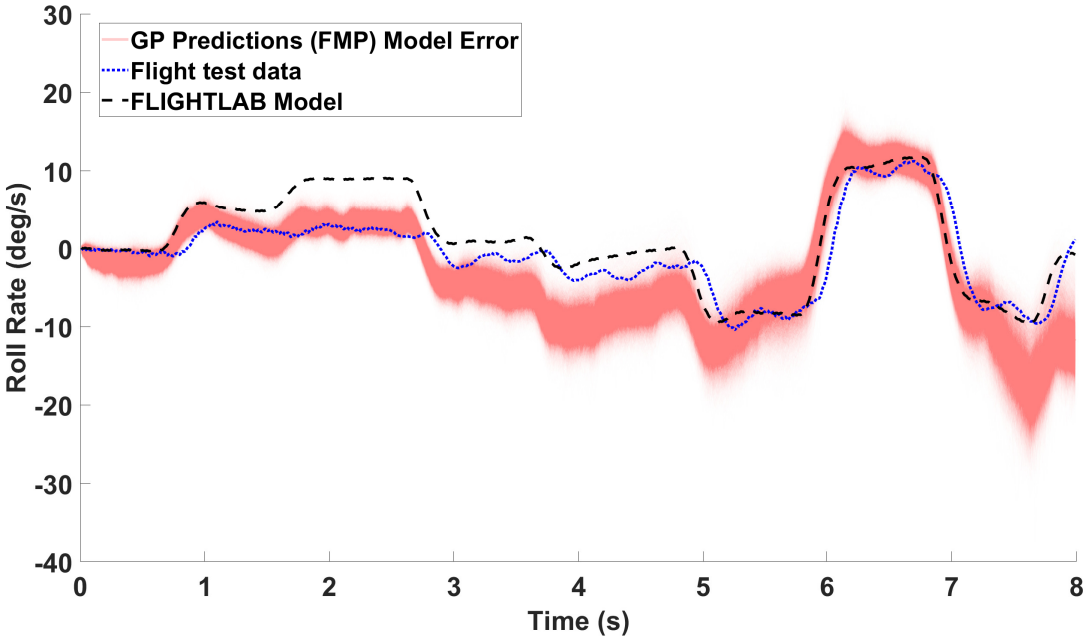


Figure I.41: Realisations of the Gaussian process roll rate prediction using the model error predictions shown in Figure I.40 with a comparison to the corresponding FLIGHTLAB model.

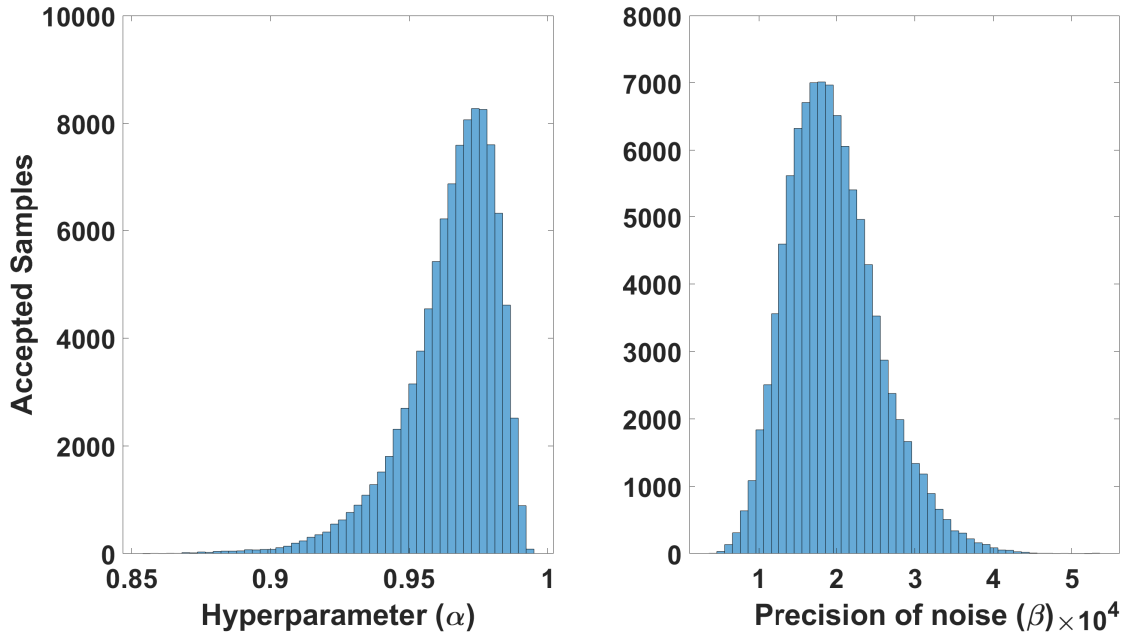


Figure I.42: Simulated annealing results used to create roll rate model error Gaussian process models, using the input structure containing the current lateral pilot input with two lagged model error roll rate terms.

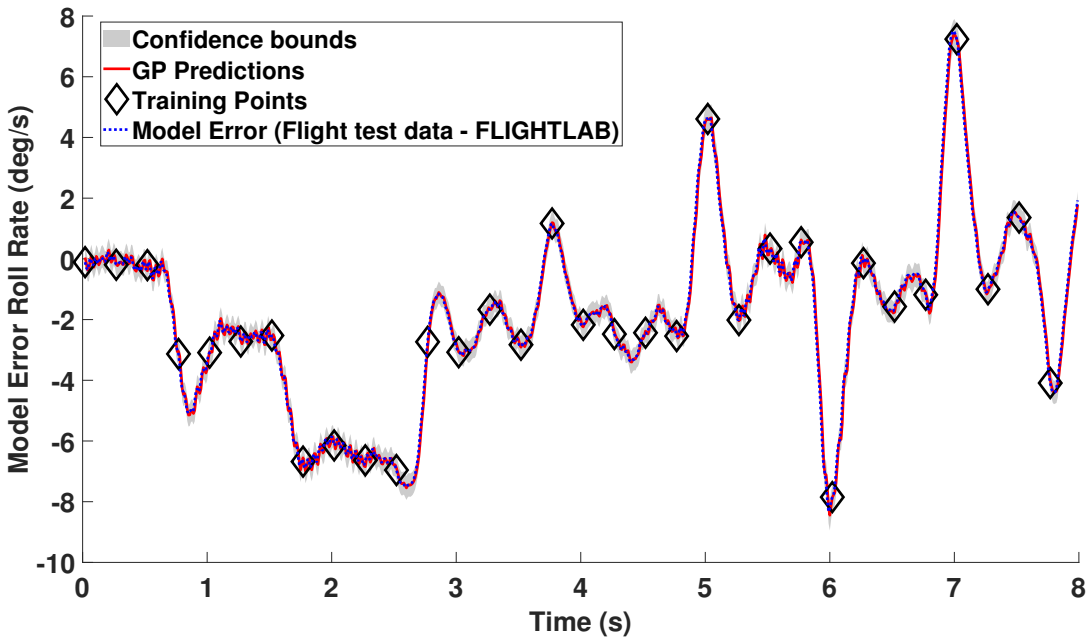


Figure I.43: Gaussian process one step ahead roll rate model error prediction using the data shown in Figure 9.9 and input structure containing the current lateral pilot input with two lagged model error roll rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.42.

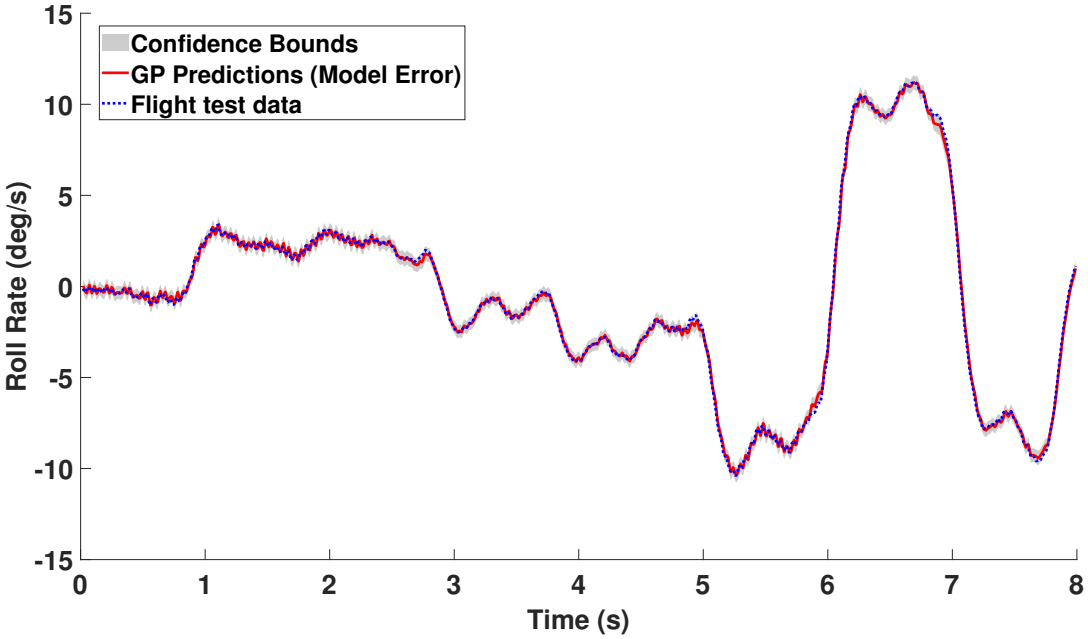


Figure I.44: Gaussian process one step ahead roll rate prediction using the model error prediction shown in Figure I.43.

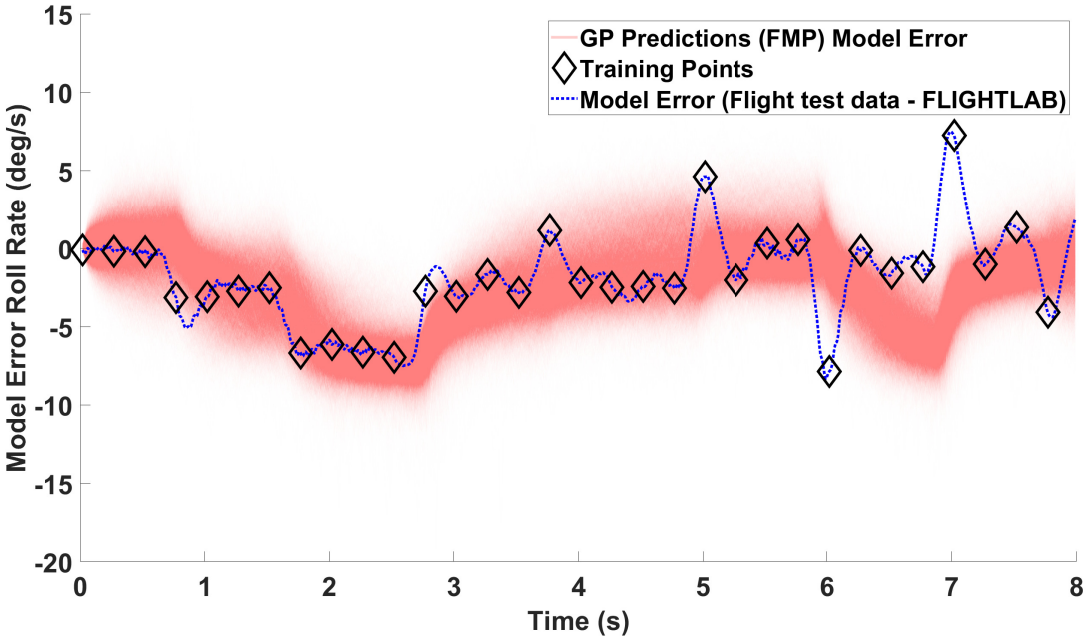


Figure I.45: Realisations of the Gaussian process roll rate model error full model prediction using the data shown in Figure 9.9 and input structure containing the current lateral pilot input with two lagged model error roll rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.42.

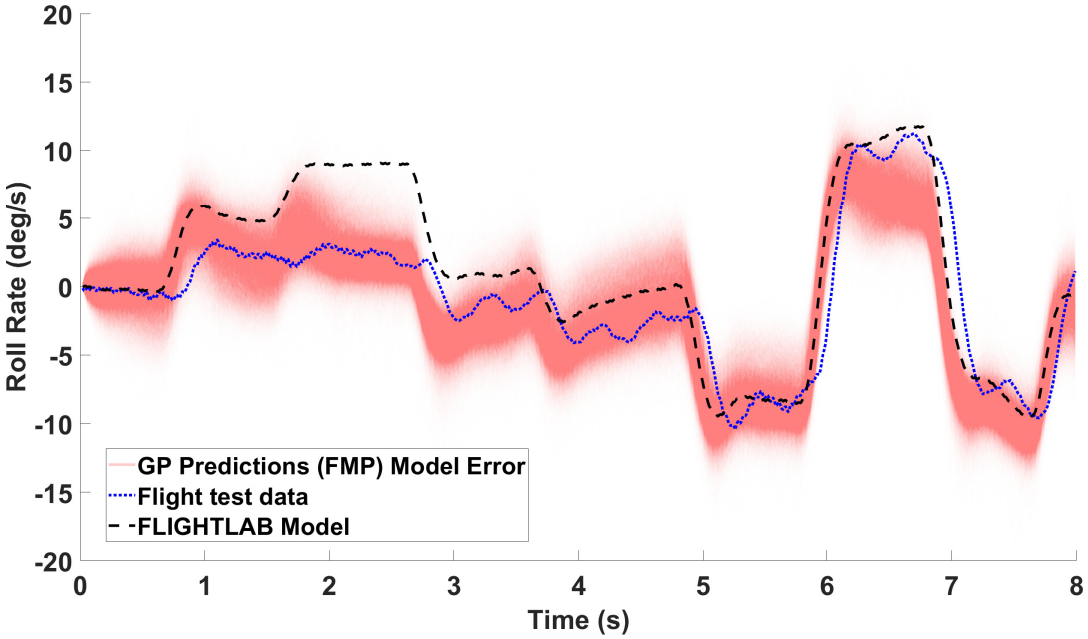


Figure I.46: Realisations of the roll rate prediction using the model error predictions shown in Figure I.45 with a comparison to the corresponding FLIGHTLAB model.

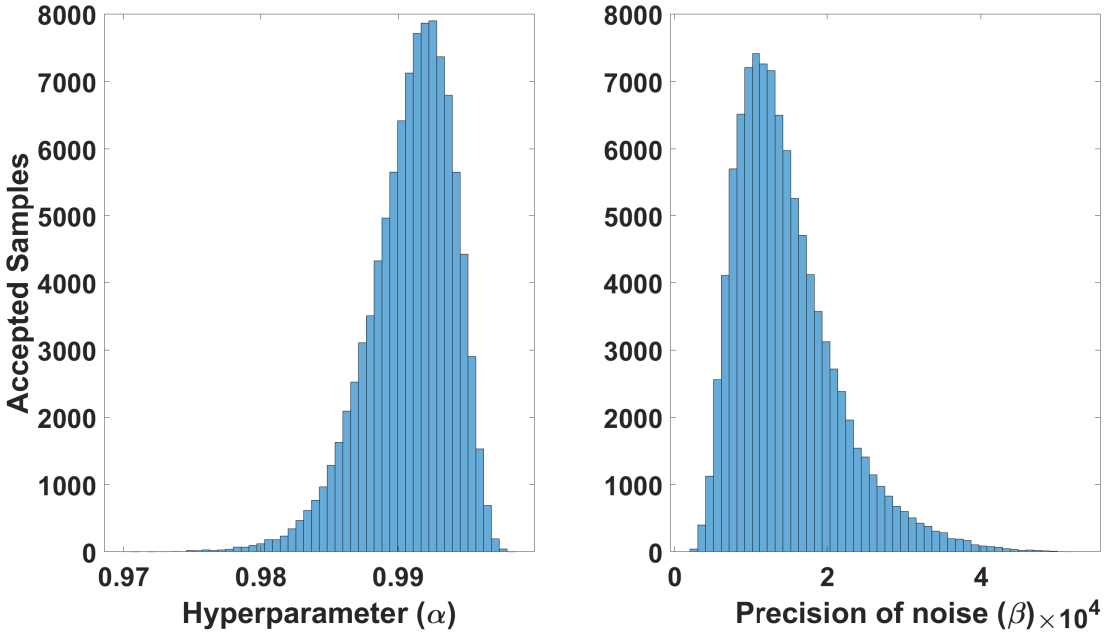


Figure I.47: Simulated annealing results used to create roll rate model error Gaussian process models, using the input structure containing all of the current pilot inputs with two lagged model error roll rate terms.

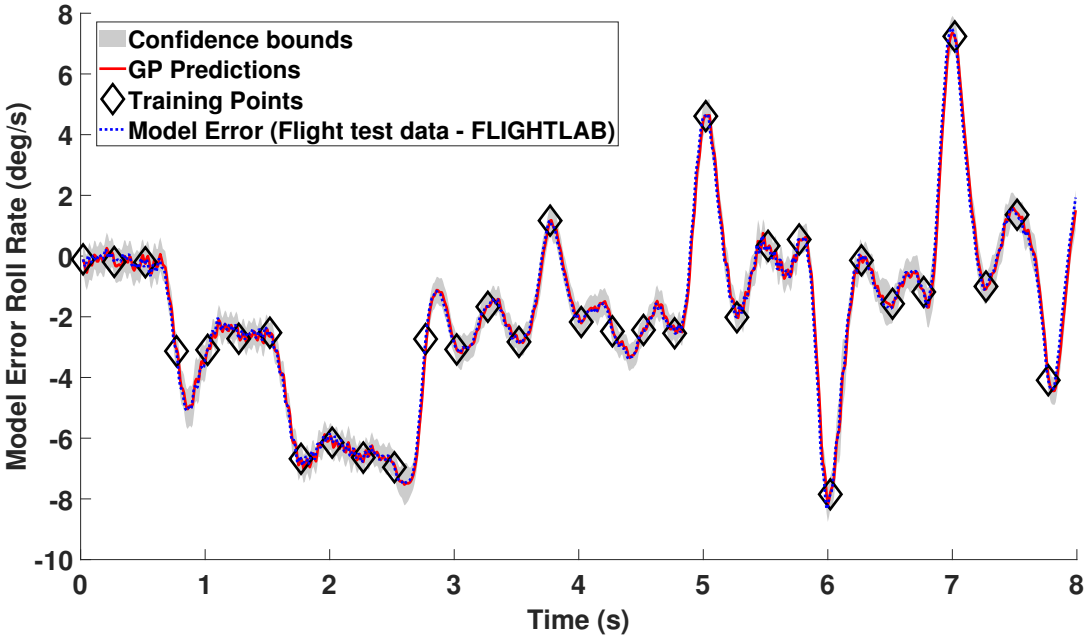


Figure I.48: Gaussian process one step ahead roll rate model error prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with two lagged model error roll rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.47.

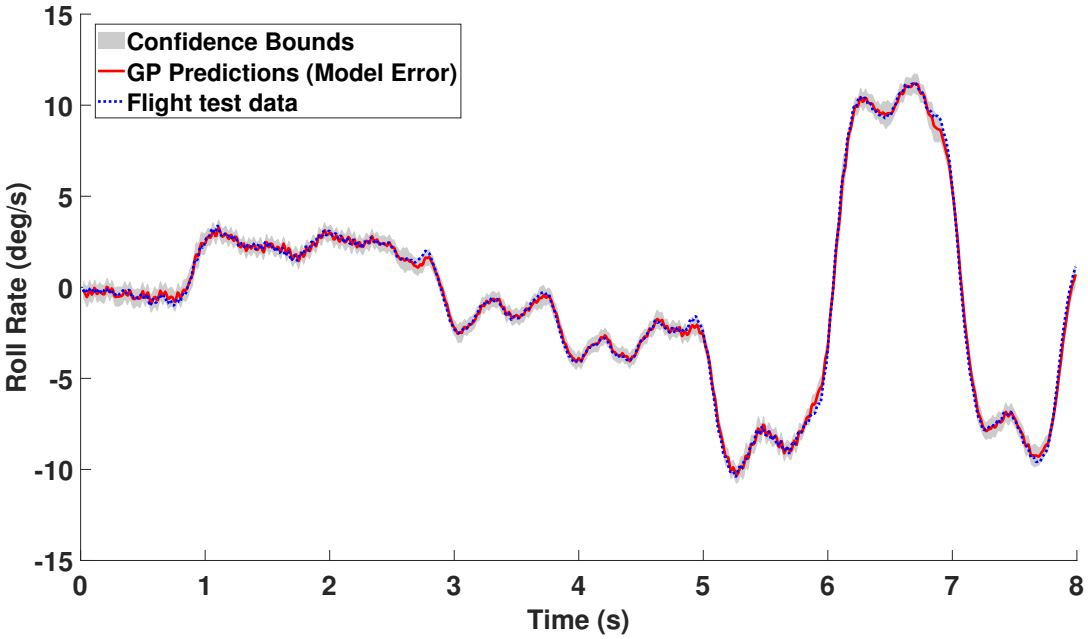


Figure I.49: Gaussian process one step ahead roll rate prediction using the model error prediction shown in Figure I.48.

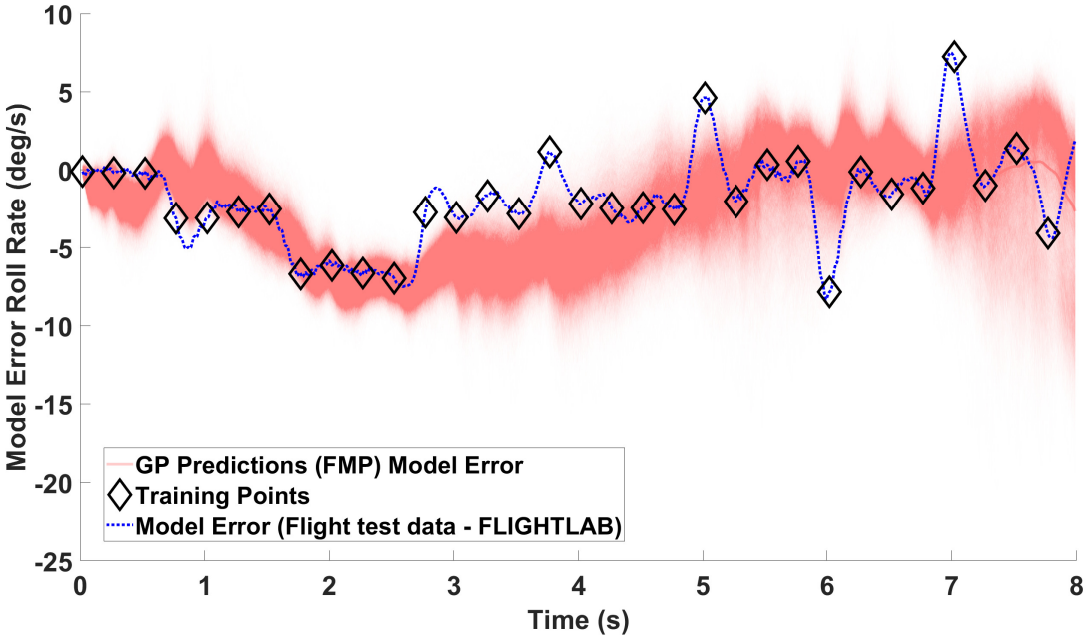


Figure I.50: Realisations of the Gaussian process roll rate model error full model prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with two lagged model error roll rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.47.

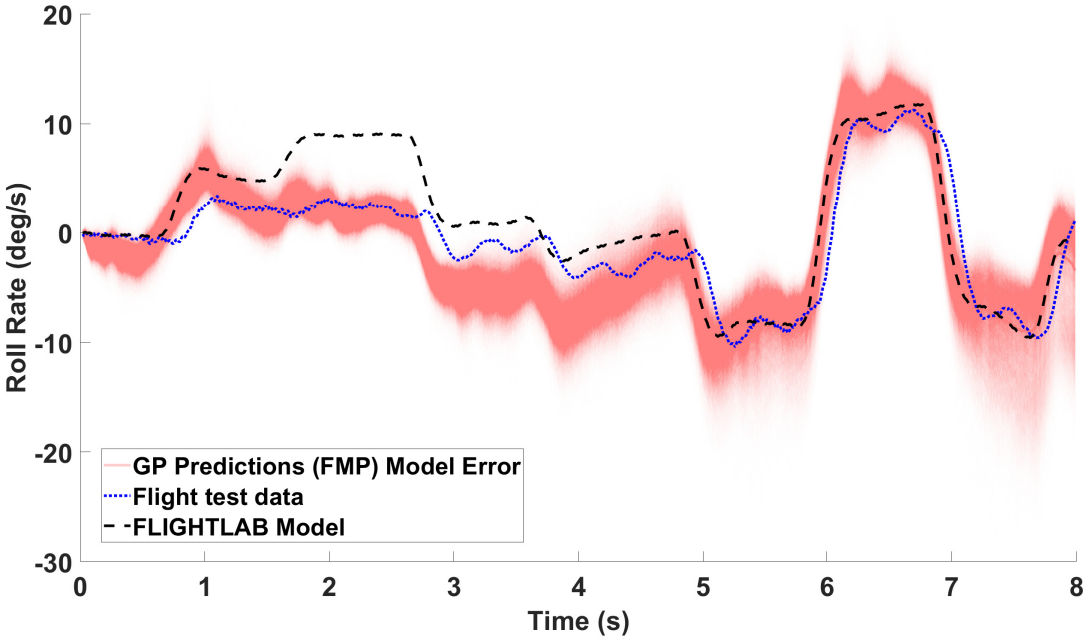


Figure I.51: Realisations of the Gaussian process roll rate prediction using the model error predictions shown in Figure I.50 with a comparison to the corresponding FLIGHTLAB model.

I.5 Yaw Rate

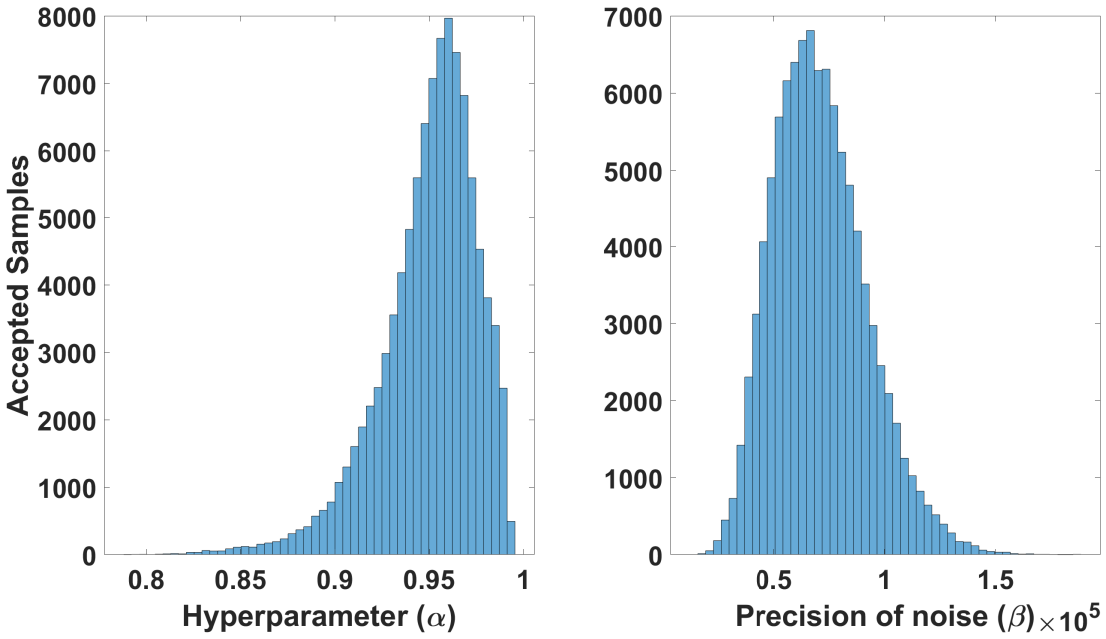


Figure I.52: Simulated annealing results used to create yaw rate model error Gaussian process models, using the input structure containing the current pedal pilot input with one lagged model error yaw rate term.

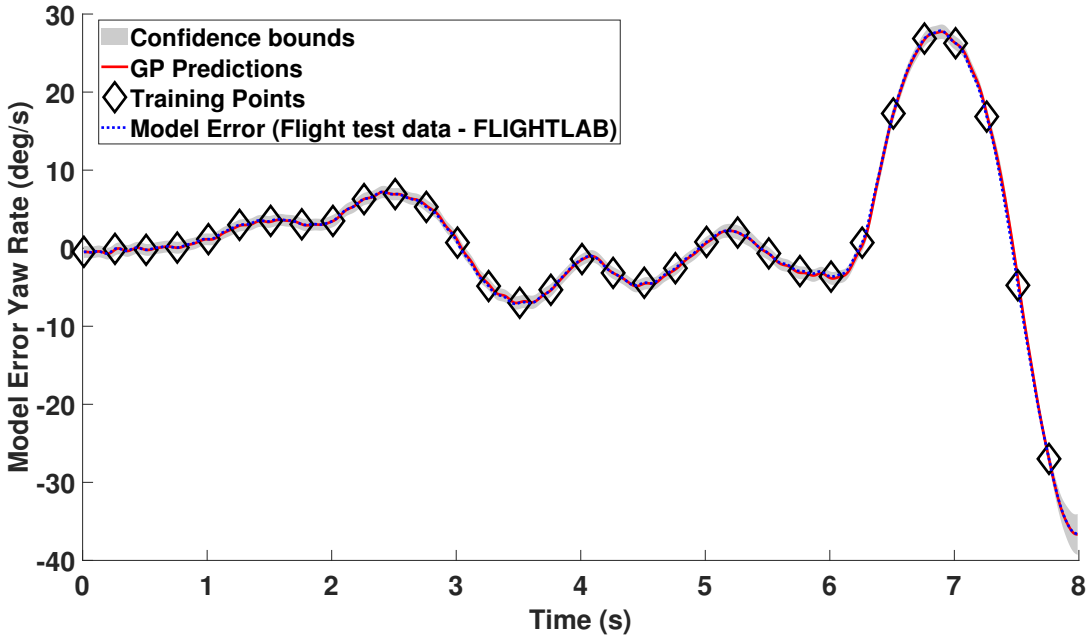


Figure I.53: Gaussian process one step ahead yaw rate model error prediction using the data shown in Figure 9.13 and input structure containing the current pedal pilot input with one lagged model error yaw rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.52.

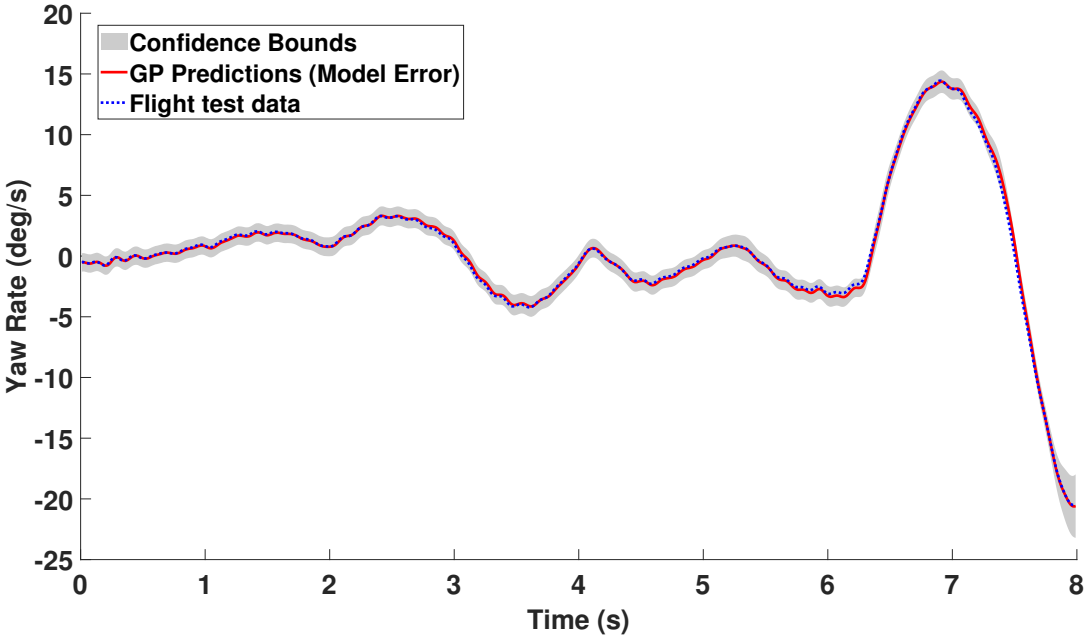


Figure I.54: Gaussian process one step ahead yaw rate prediction using the model error prediction shown in Figure I.53.

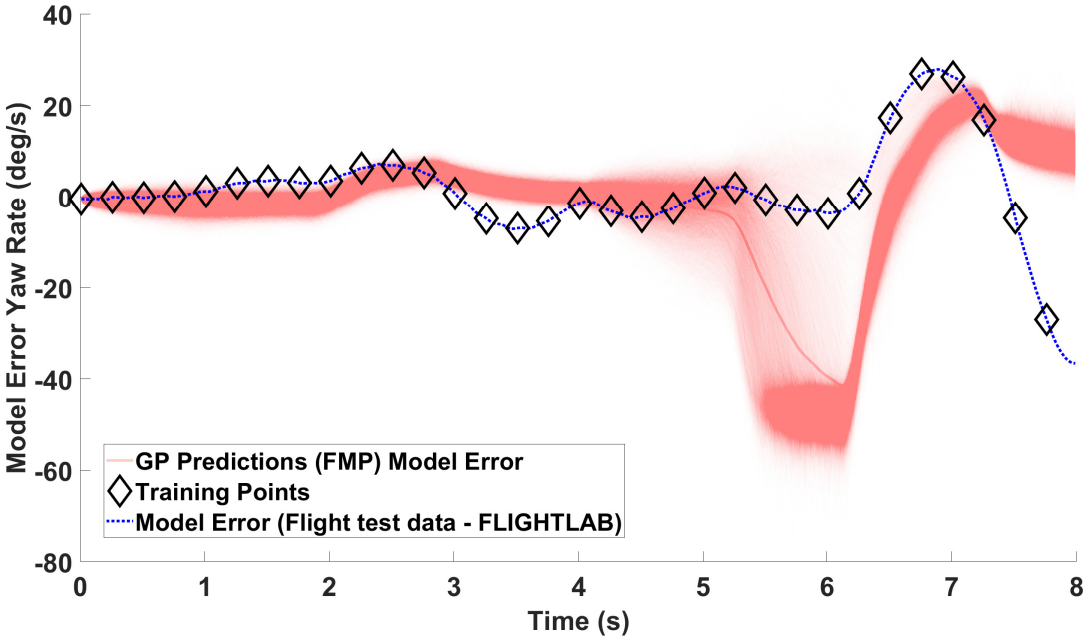


Figure I.55: Realisations of the Gaussian process yaw rate model error full model prediction using the data shown in Figure 9.13 and input structure containing the current pedal pilot input with one lagged model error yaw rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.52.

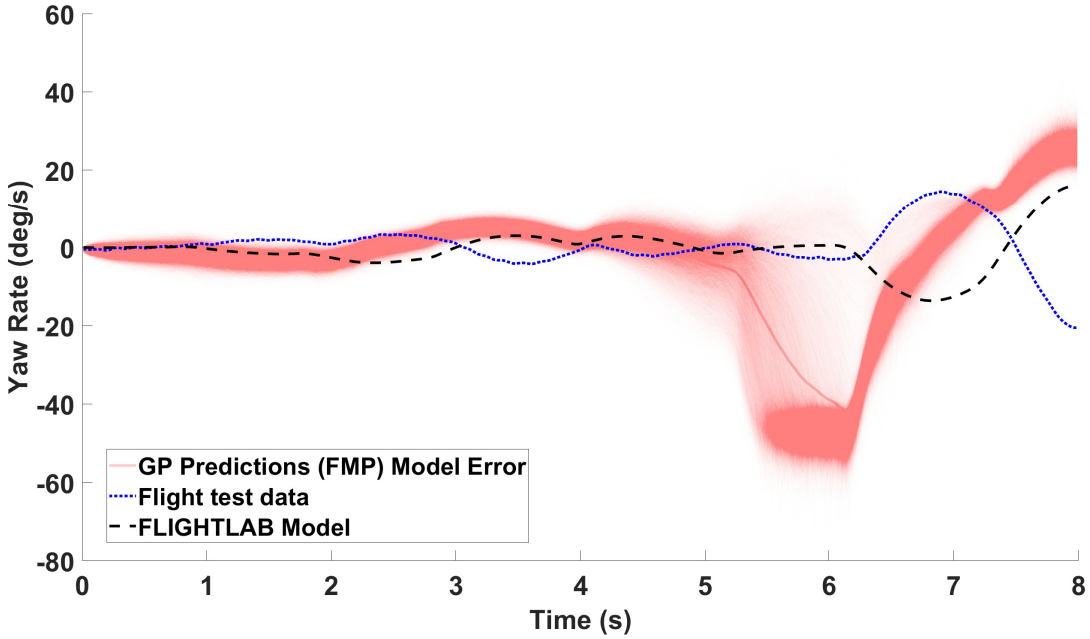


Figure I.56: Realisations of the yaw rate prediction using the model error predictions shown in Figure I.55 with a comparison to the corresponding FLIGHTLAB model.

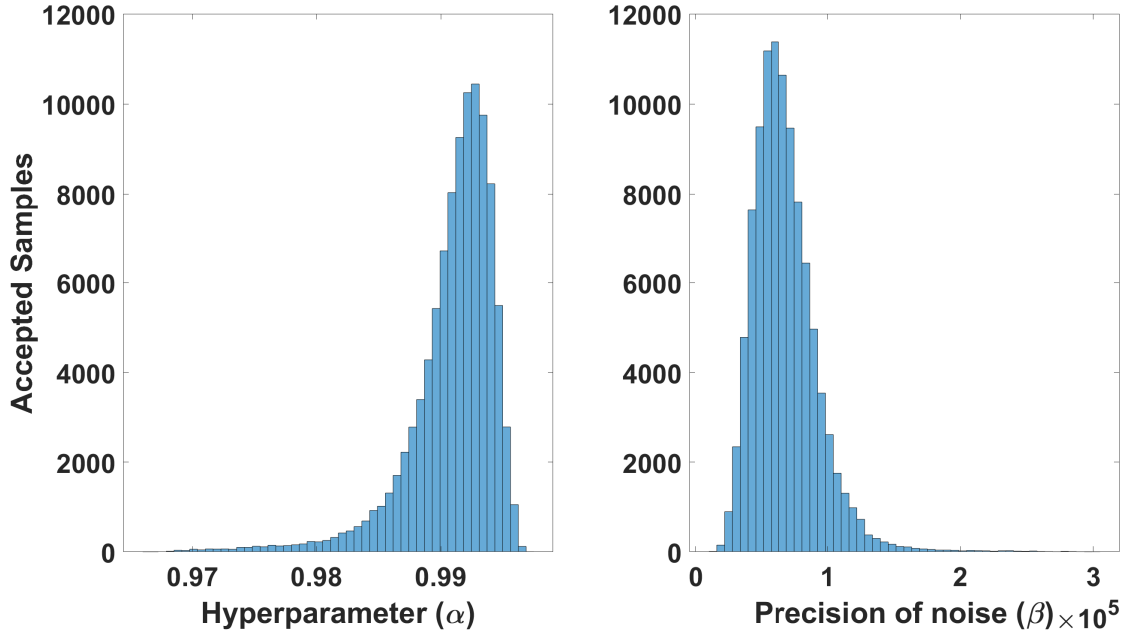


Figure I.57: Simulated annealing results used to create yaw rate model error Gaussian process models, using the input structure containing all of the current pilot inputs with one lagged model error yaw rate term.

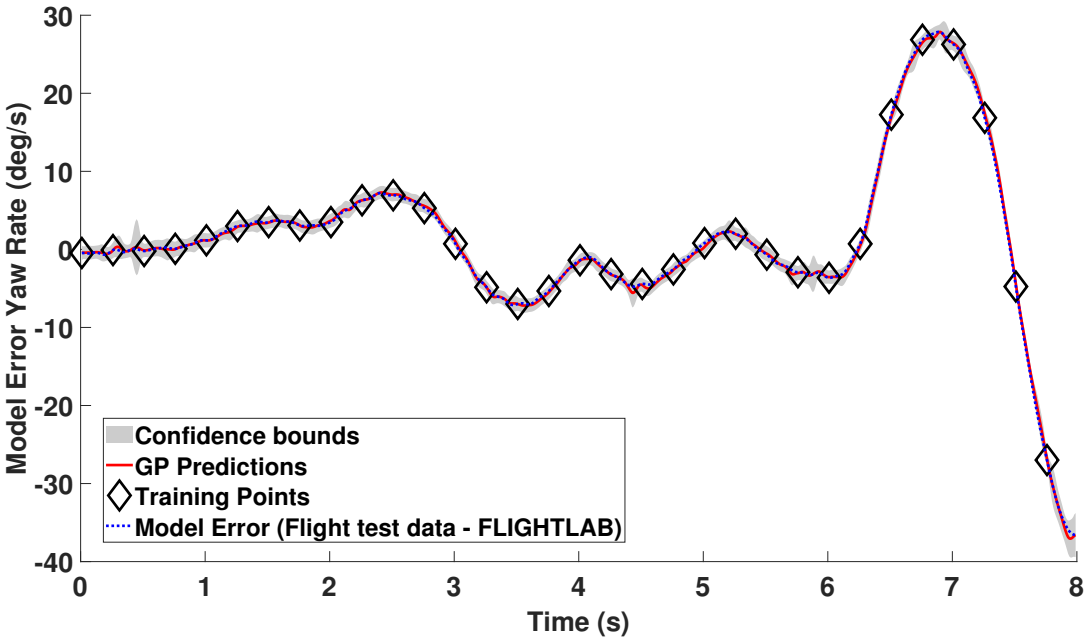


Figure I.58: Gaussian process one step ahead yaw rate model error prediction using the data shown in Figure 9.13 and input structure containing all of the current pilot inputs with one lagged model error yaw rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.57.

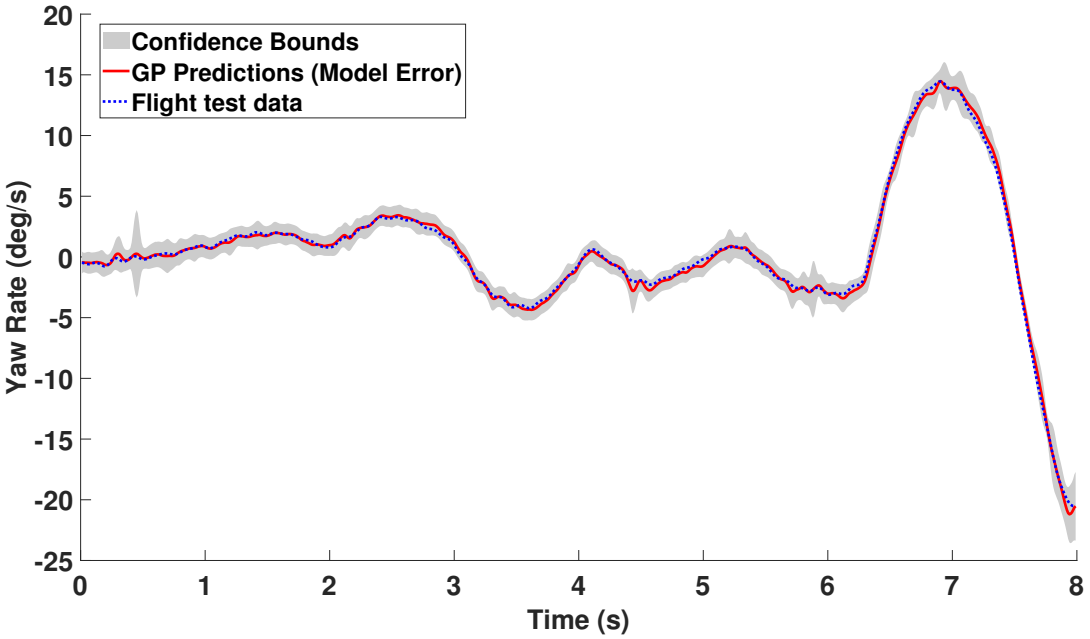


Figure I.59: Gaussian process one step ahead yaw rate prediction using the model error prediction shown in Figure I.58.

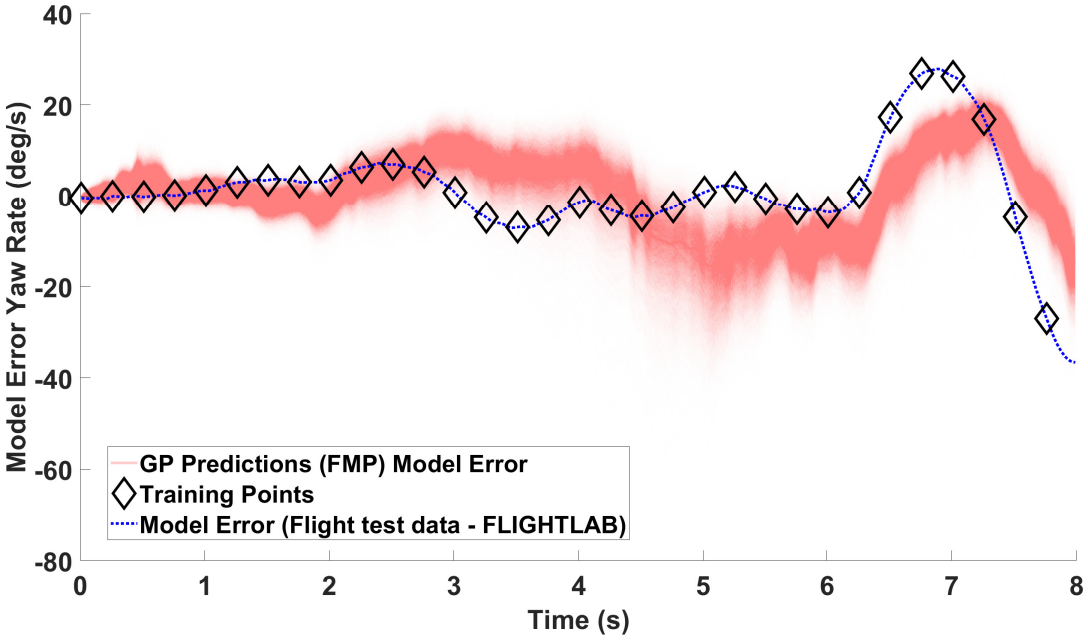


Figure I.60: Realisations of the Gaussian process yaw rate model error full model prediction using the data shown in Figure 9.13 and input structure containing all of the current pilot inputs with one lagged model error yaw rate term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.57.

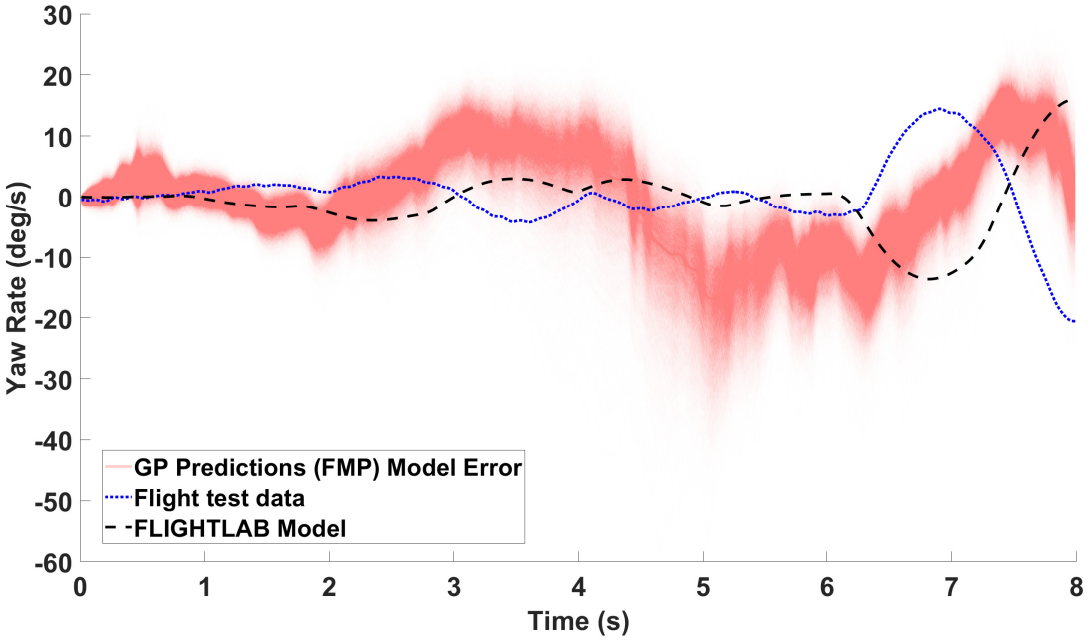


Figure I.61: Realisations of the yaw rate prediction using the model error predictions shown in Figure I.60 with a comparison to the corresponding FLIGHTLAB model.

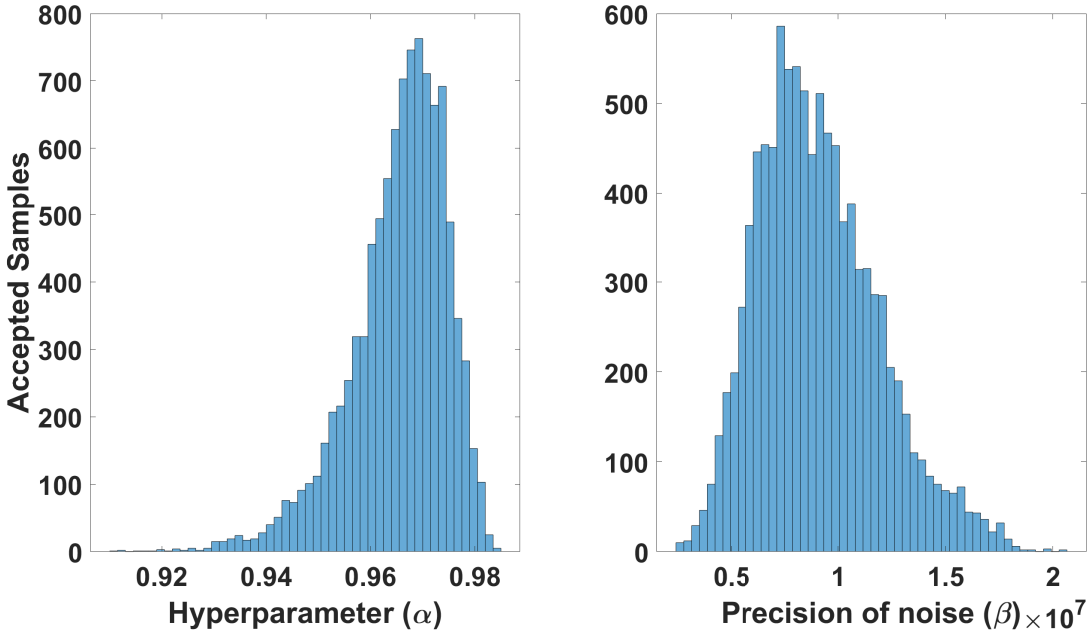


Figure I.62: Simulated annealing results used to create yaw rate model error Gaussian process models, using the input structure containing the current pedal pilot input with two lagged model error yaw rate terms.

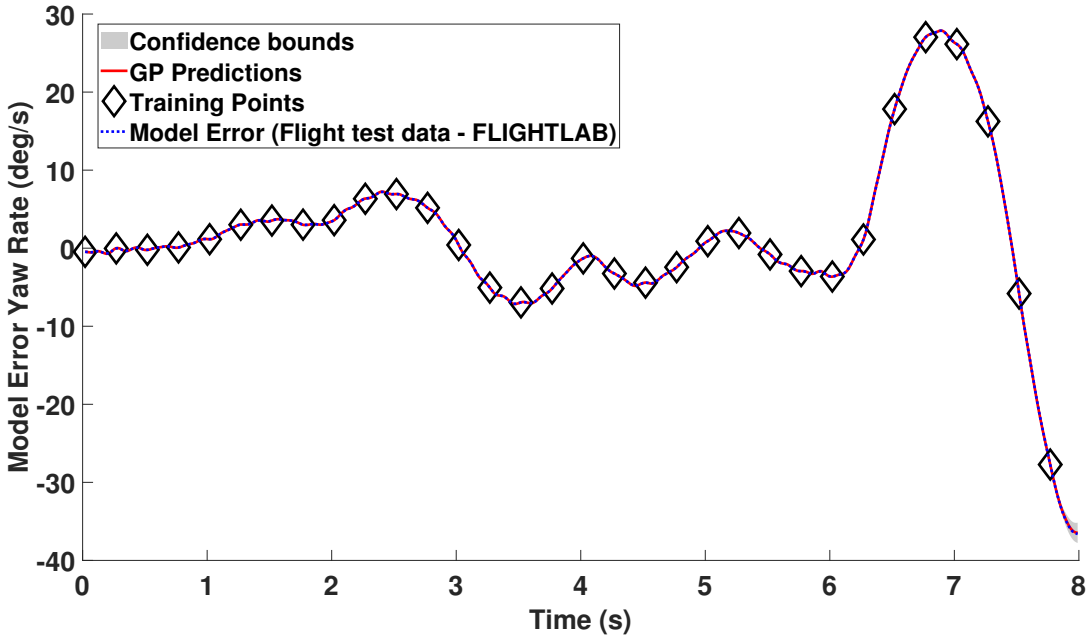


Figure I.63: Gaussian process one step ahead yaw rate model error prediction using the data shown in Figure 9.13 and input structure containing the current pedal pilot input with two lagged model error yaw rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.62.

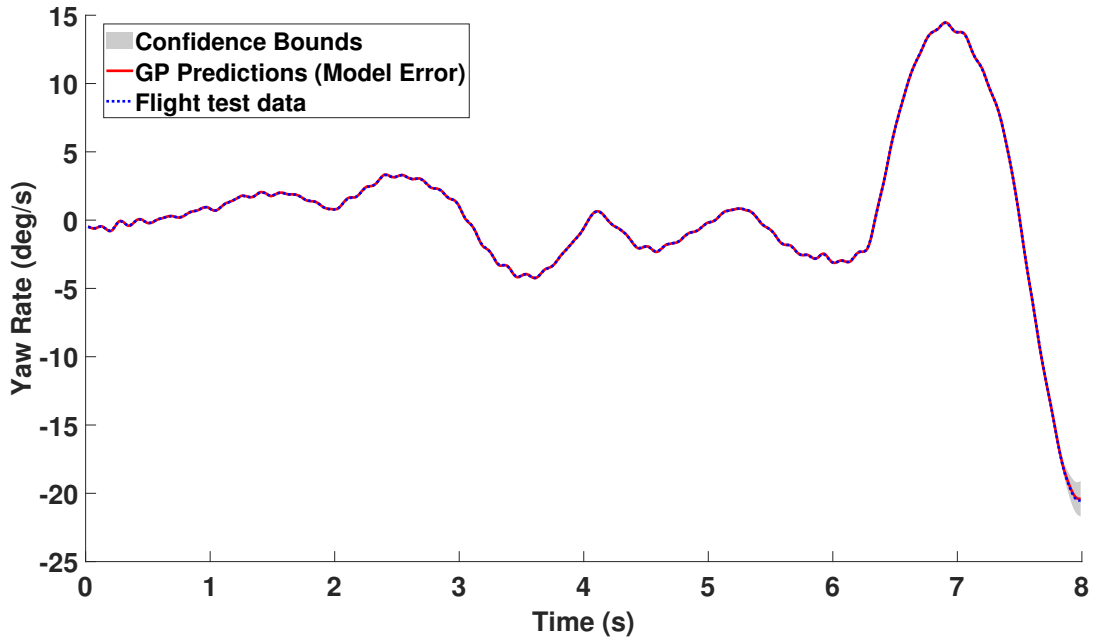


Figure I.64: Gaussian process one step ahead yaw rate prediction using the model error prediction shown in Figure I.63.

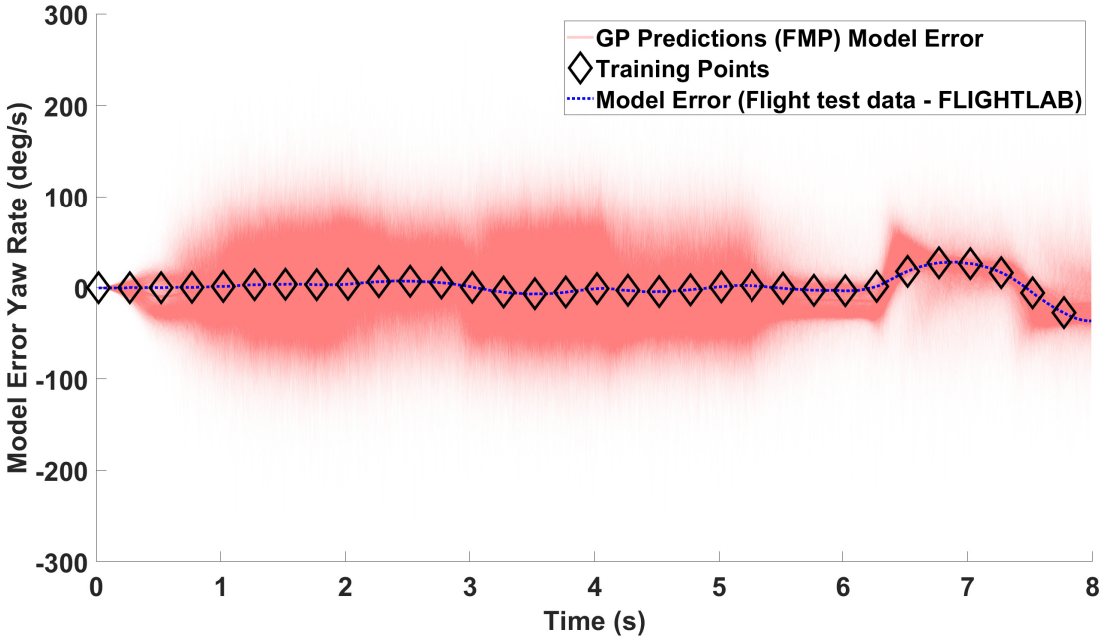


Figure I.65: Realisations of the Gaussian process yaw rate model error full model prediction using the data shown in Figure 9.13 and input structure containing the current pedal pilot input with two lagged model error yaw rates terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.62.

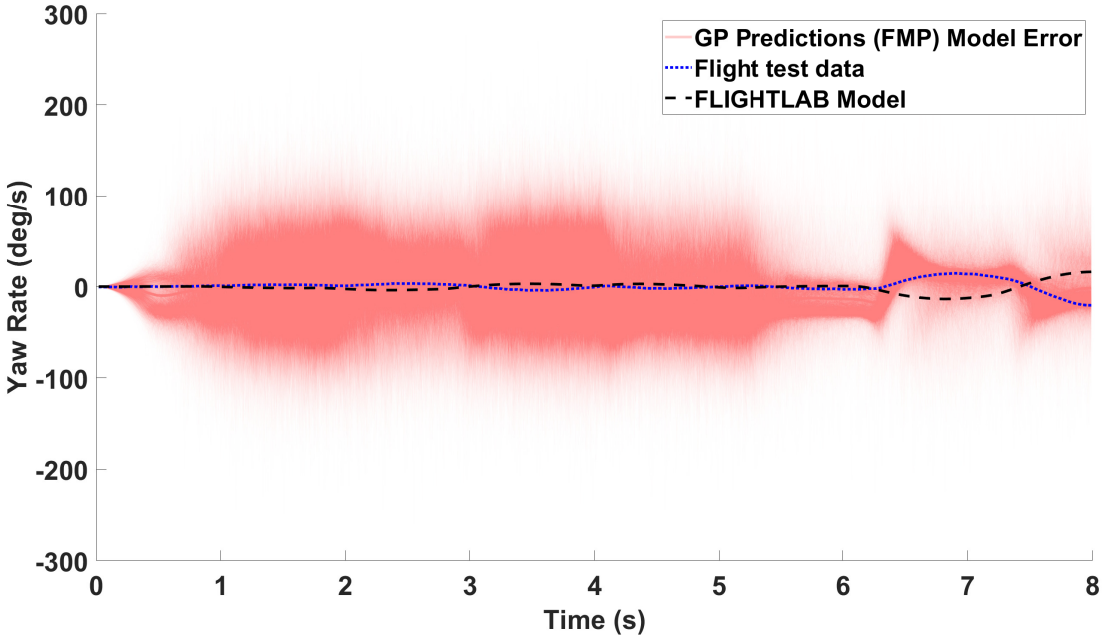


Figure I.66: Realisations of the yaw rate prediction using the model error predictions shown in Figure I.65 with a comparison to the corresponding FLIGHTLAB model.

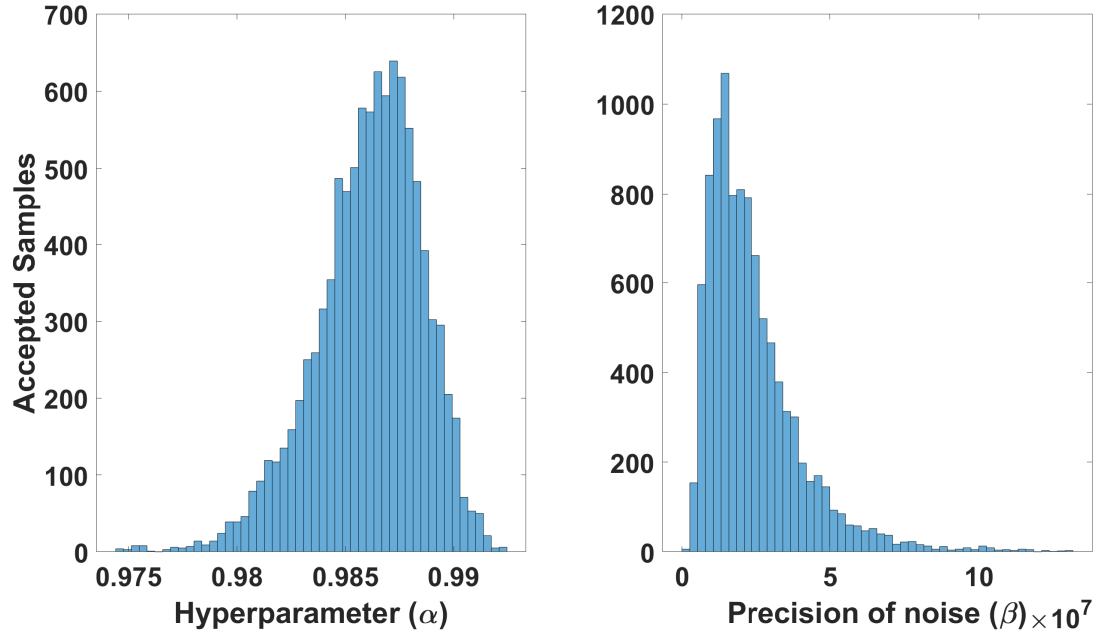


Figure I.67: Simulated annealing results used to create yaw rate model error Gaussian process models, using the input structure containing all of the current pilot inputs with two lagged model error yaw rate terms.

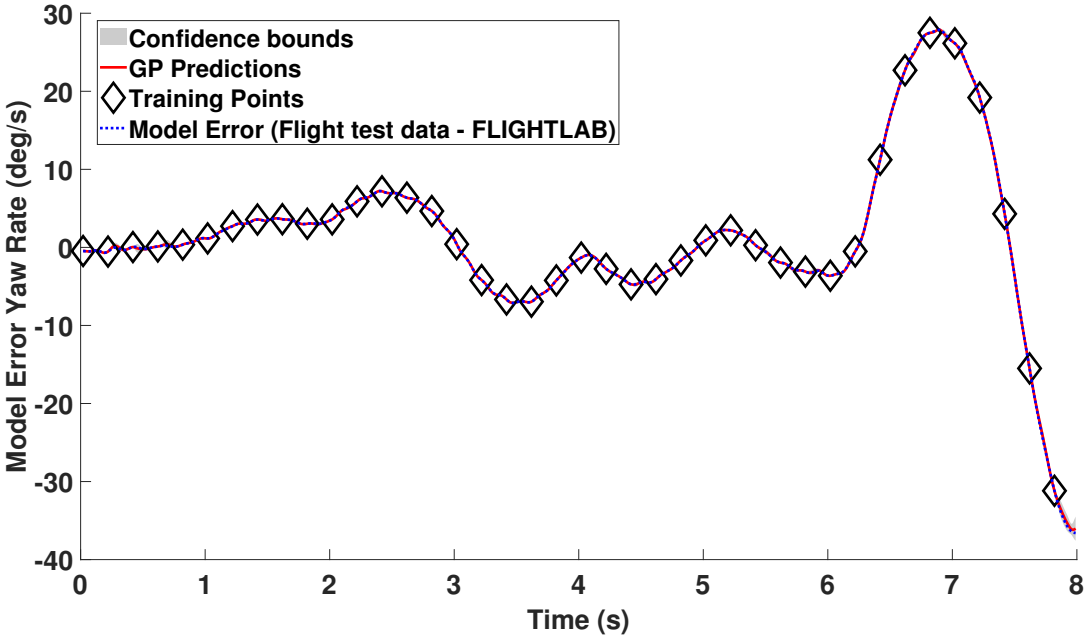


Figure I.68: Gaussian process one step ahead yaw rate model error prediction using the data shown in Figure 9.13 and input structure containing all of the current pilot inputs with two lagged model error yaw rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.67.

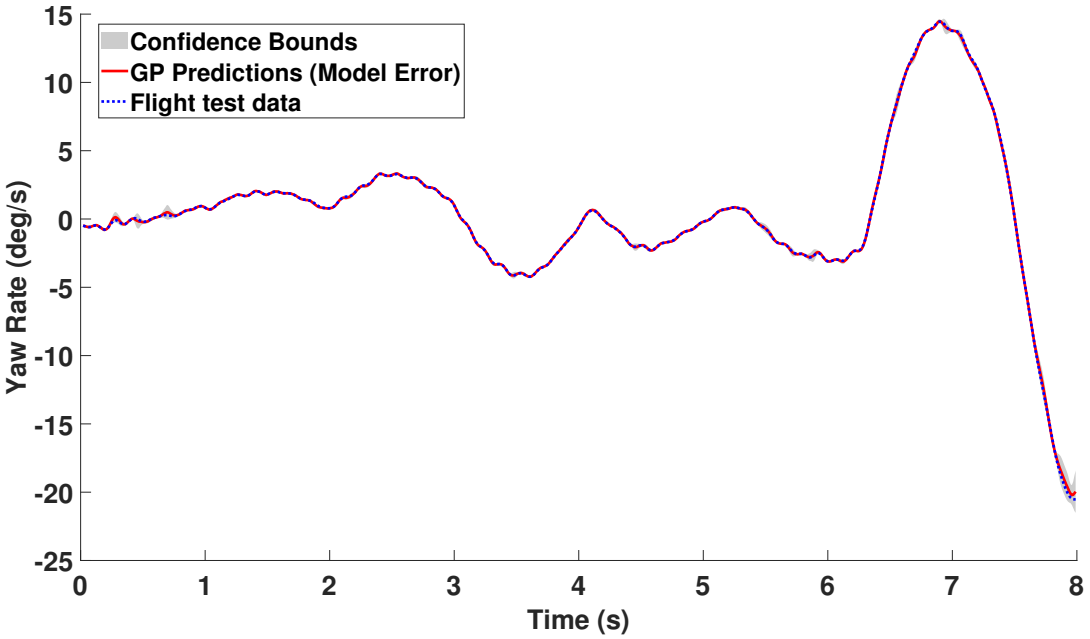


Figure I.69: Gaussian process one step ahead yaw rate prediction using the model error prediction shown in Figure I.68.

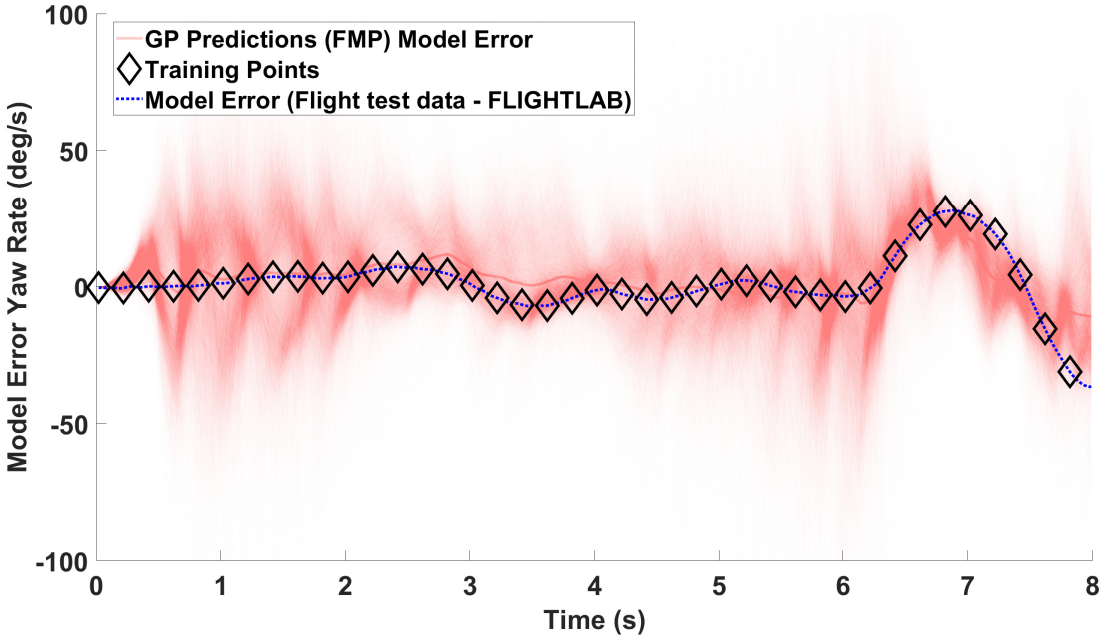


Figure I.70: Realisations of the Gaussian process yaw rate model error full model prediction using the data shown in Figure 9.13 and input structure containing all of the current pilot inputs with two lagged model error yaw rate terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.67.

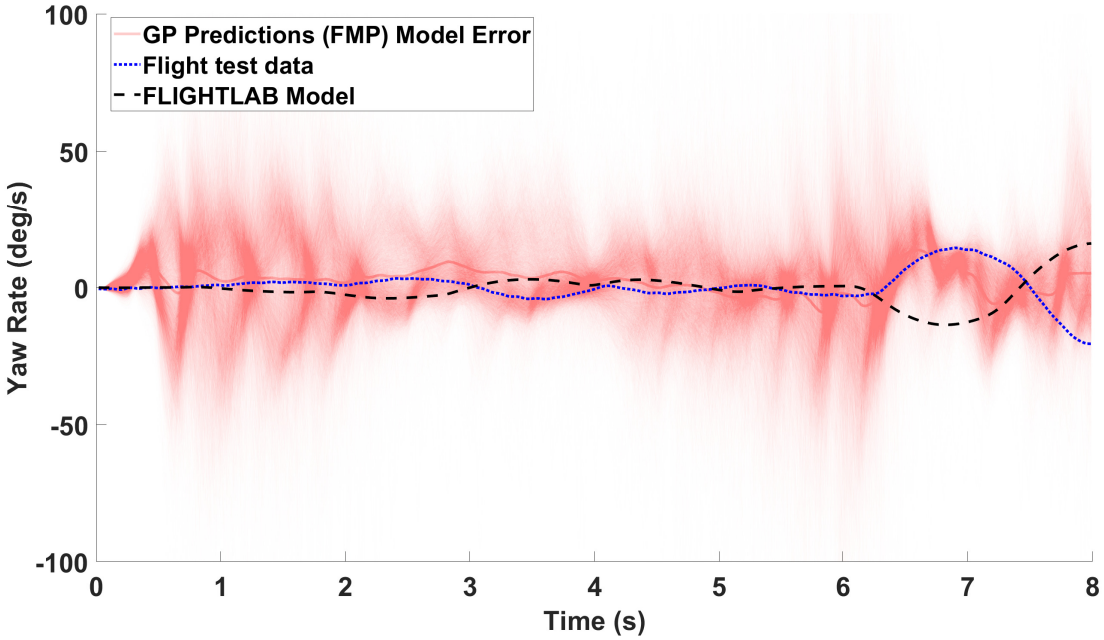


Figure I.71: Realisations of the yaw rate prediction using the model error predictions shown in Figure I.70 with a comparison to the corresponding FLIGHTLAB model.

I.6 Heave

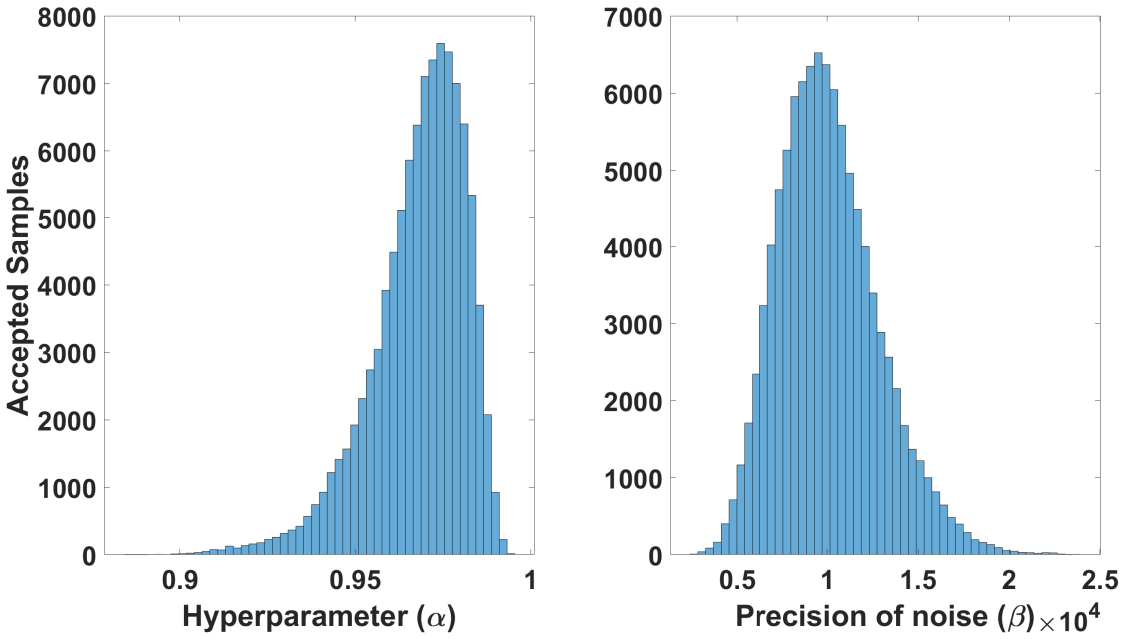


Figure I.72: Simulated annealing results used to create heave model error Gaussian process models, using the input structure containing the current collective pilot input with one lagged model error heave term.

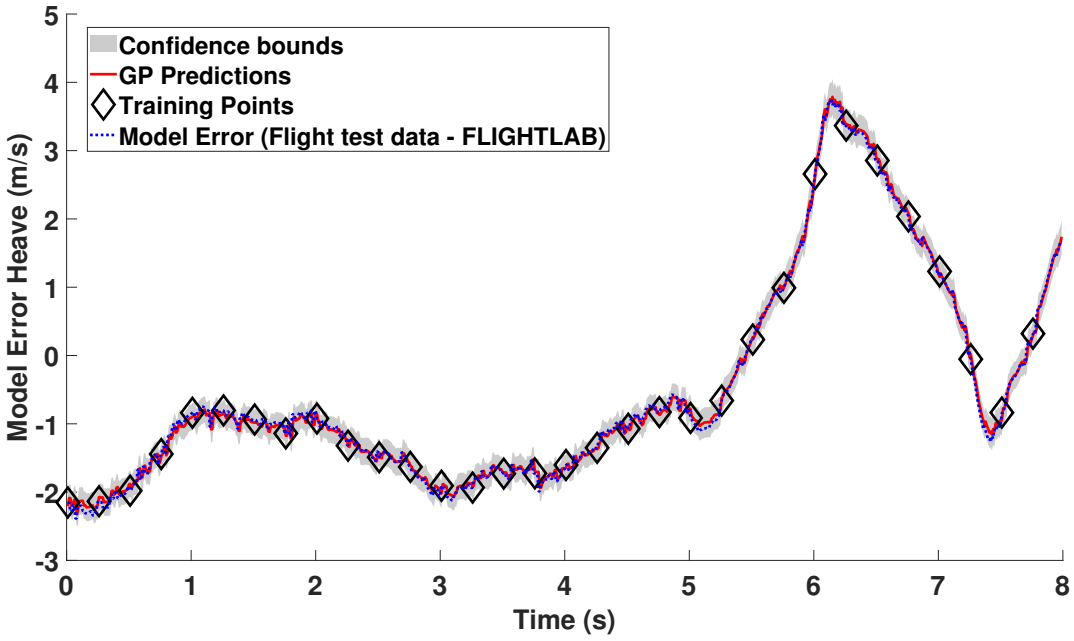


Figure I.73: Gaussian process one step ahead heave model error prediction using the data shown in Figure 9.17 and input structure containing the current collective pilot input with one lagged model error heave term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.72.

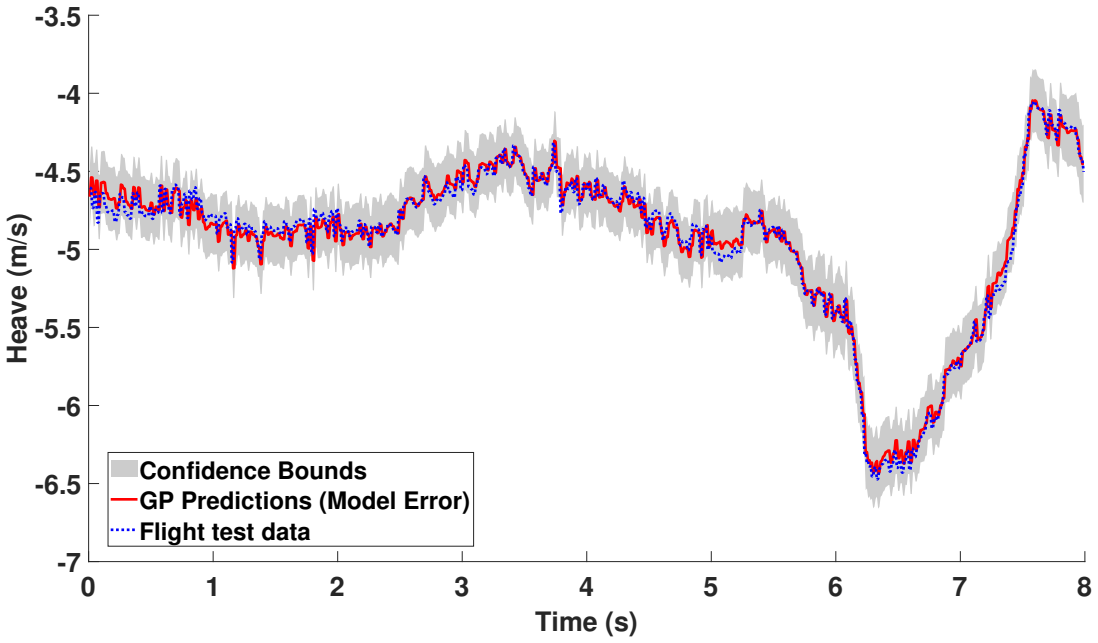


Figure I.74: Gaussian process one step ahead heave prediction using the model error prediction shown in Figure I.73.

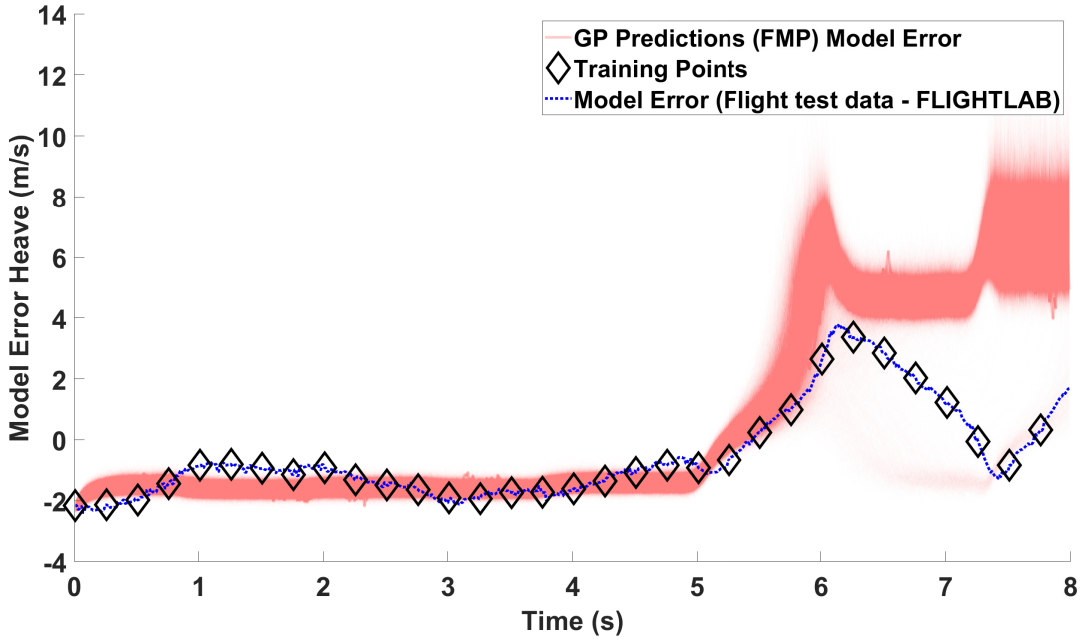


Figure I.75: Realisations of the Gaussian process heave model error full model prediction using the data shown in Figure 9.17 and input structure containing the current collective pilot input with one lagged model error heave term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.72.

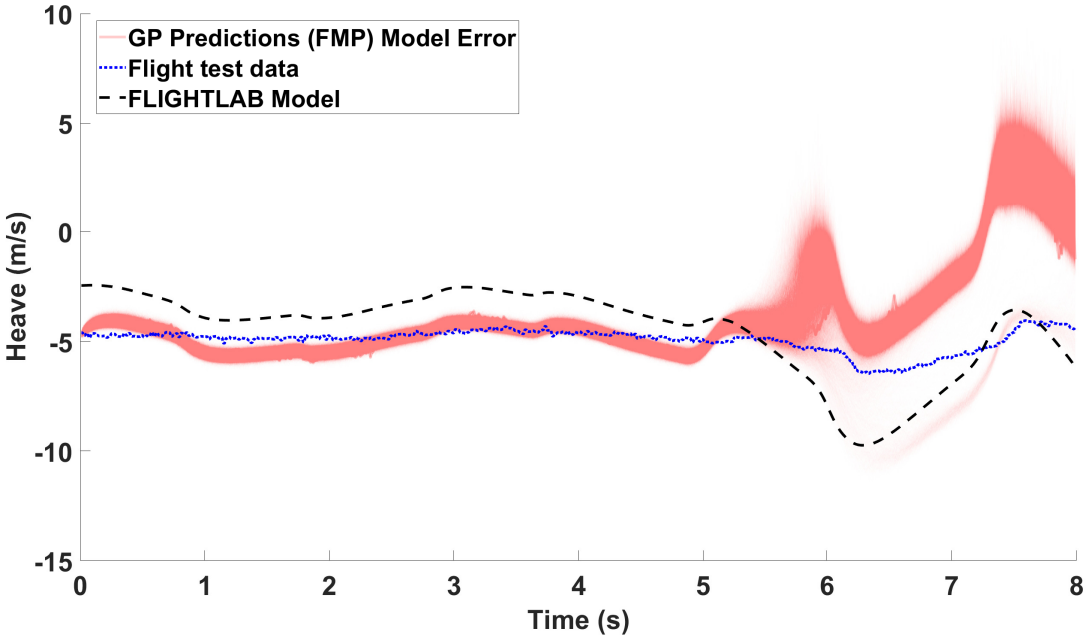


Figure I.76: Realisations of the heave prediction using the model error predictions shown in Figure I.75 with a comparison to the corresponding FLIGHTLAB model.

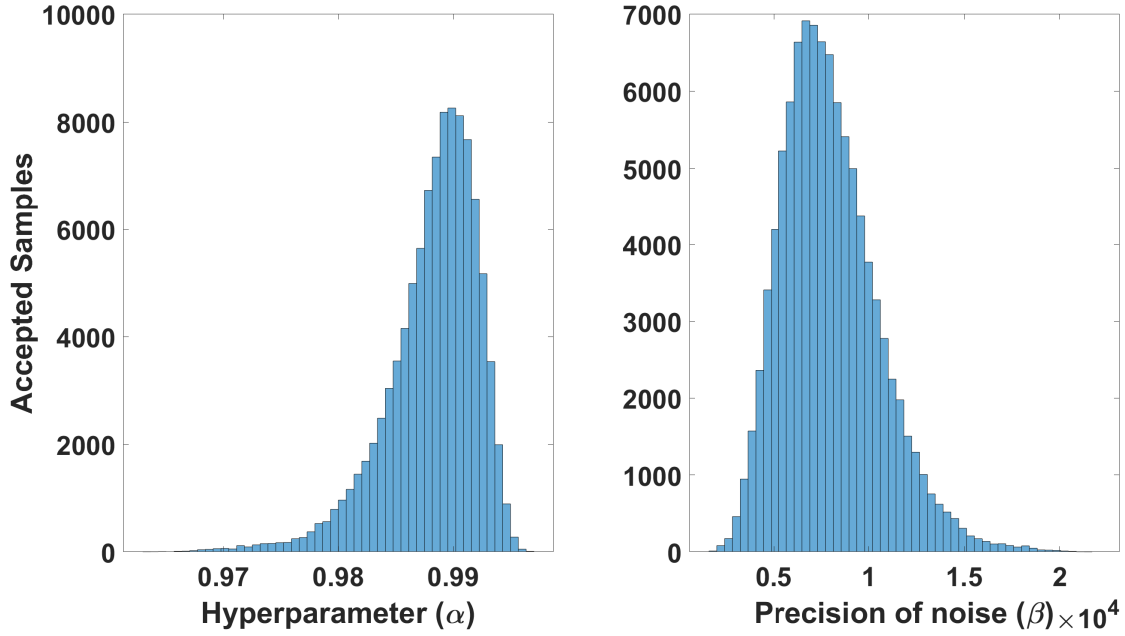


Figure I.77: Simulated annealing results used to create heave model error Gaussian process models, using the input structure containing all of the current pilot inputs with one lagged model error heave term.

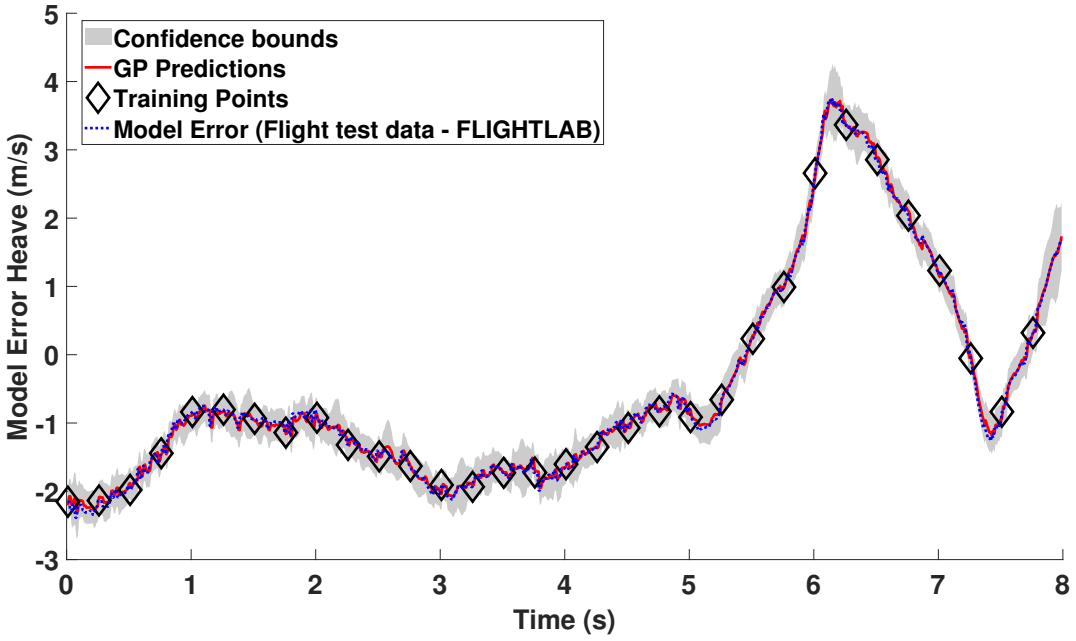


Figure I.78: Gaussian process one step ahead heave model error prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with one lagged model error heave term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.77.

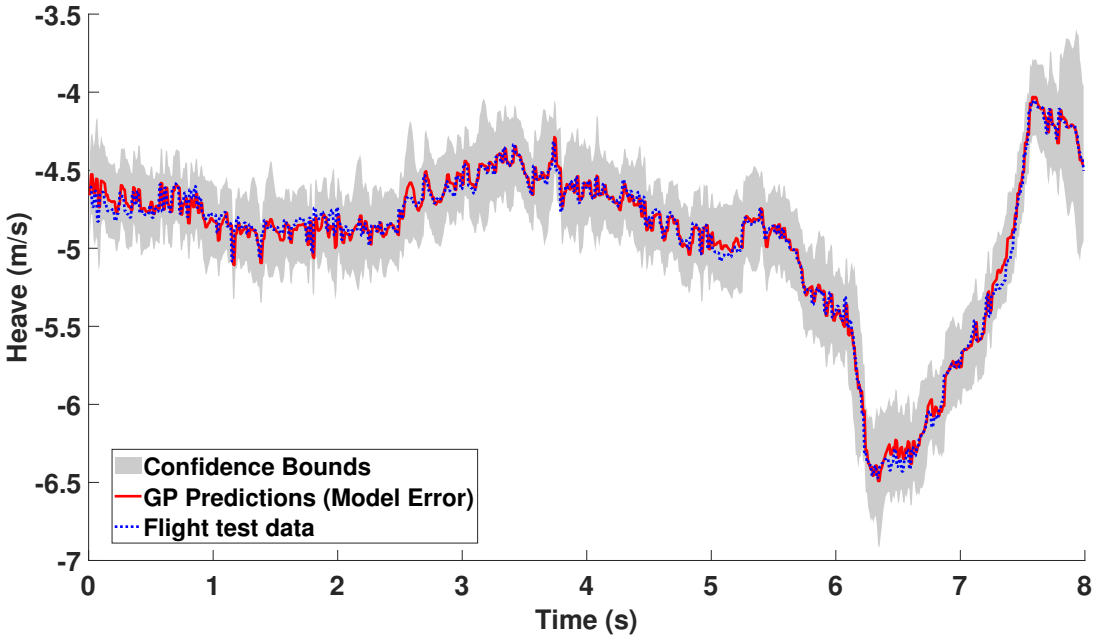


Figure I.79: Gaussian process one step ahead heave prediction using the model error prediction shown in Figure I.78.

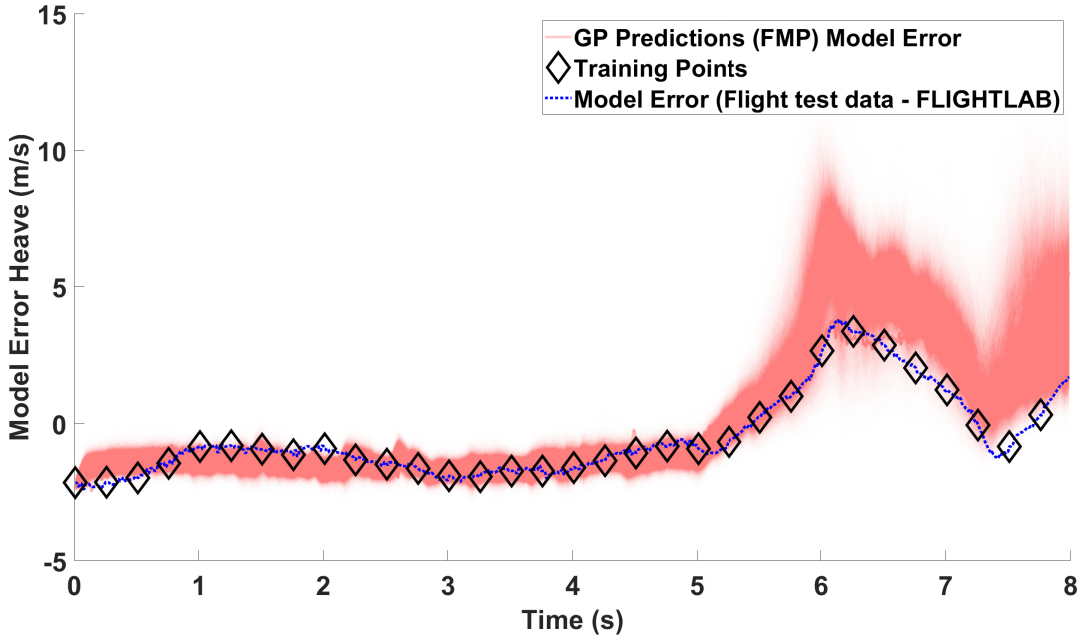


Figure I.80: Realisations of the Gaussian process model error heave full model prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with one lagged model error heave term utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.77.

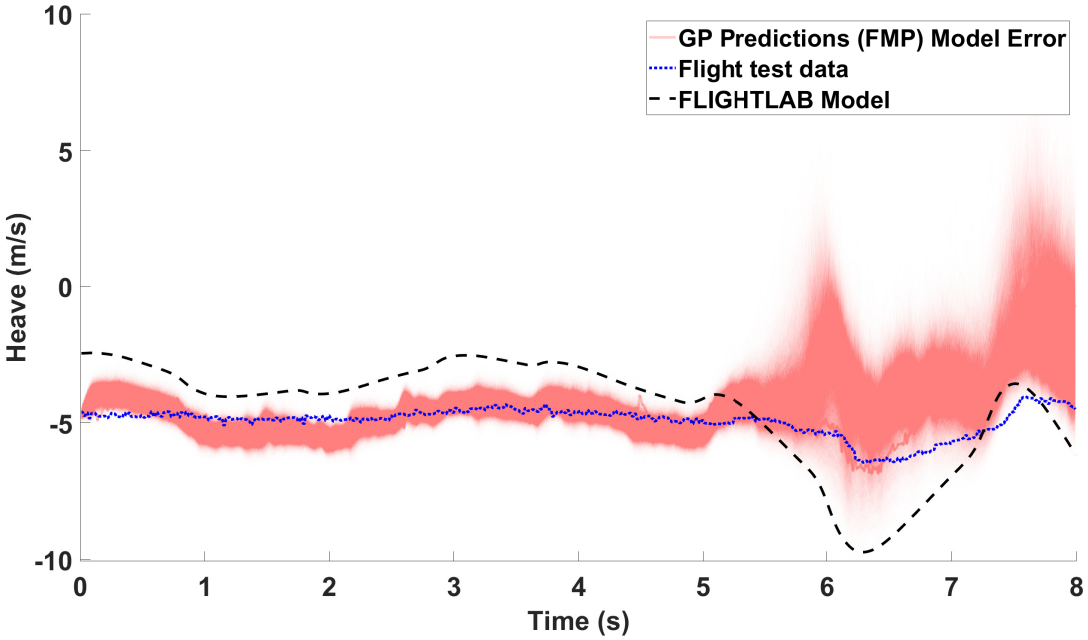


Figure I.81: Realisations of the Gaussian process heave prediction using the model error predictions shown in Figure I.80 with a comparison to the corresponding FLIGHTLAB model.

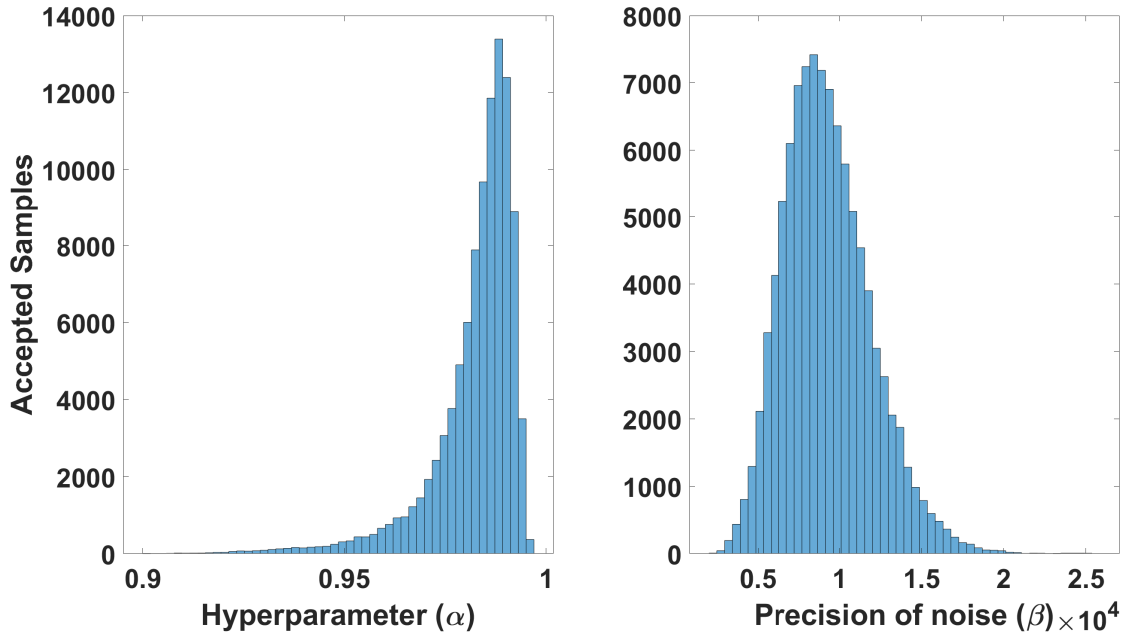


Figure I.82: Simulated annealing results used to create heave model error Gaussian process models, using the input structure containing the current collective pilot input with two lagged model error heave terms.

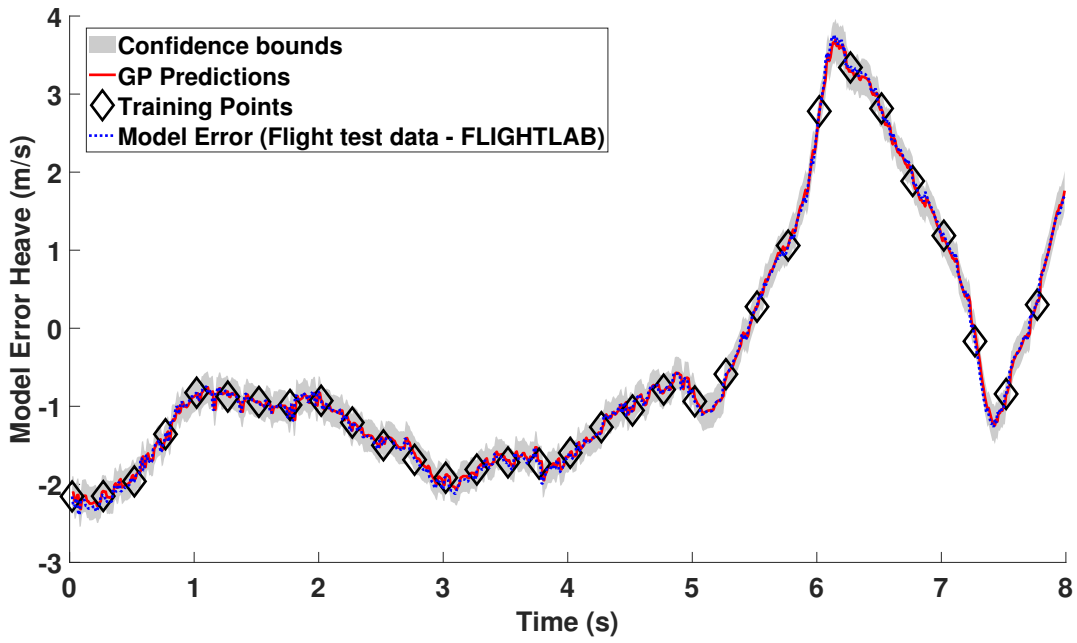


Figure I.83: Gaussian process one step ahead heave model error prediction using the data shown in Figure 9.17 and input structure containing the current collective pilot input with two lagged model error heave terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.82.

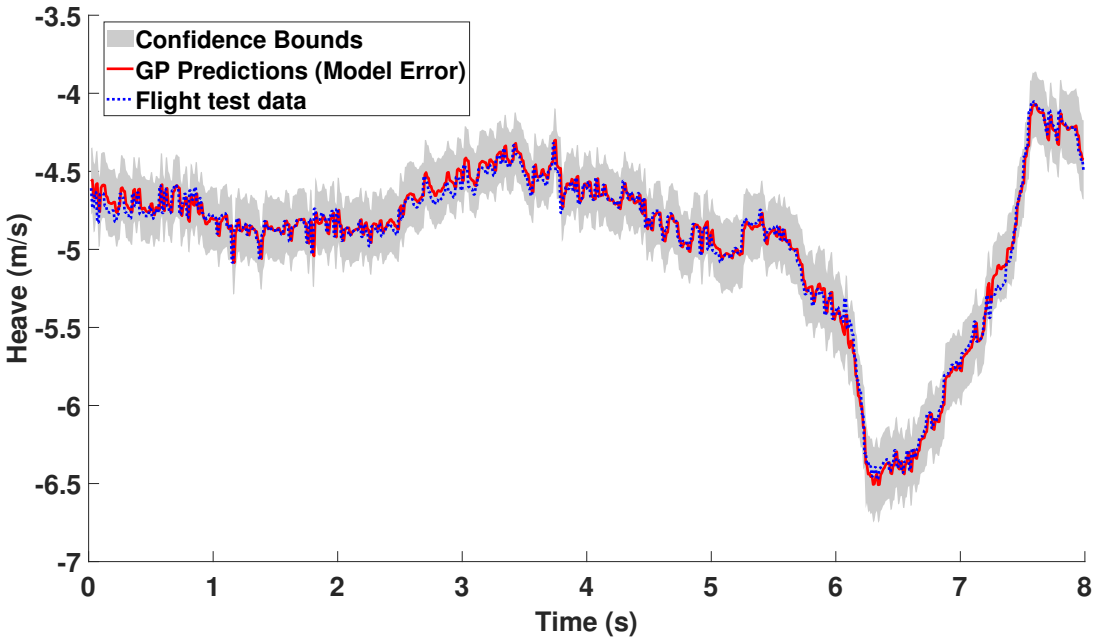


Figure I.84: Gaussian process one step ahead heave prediction using the model error prediction shown in Figure I.83.

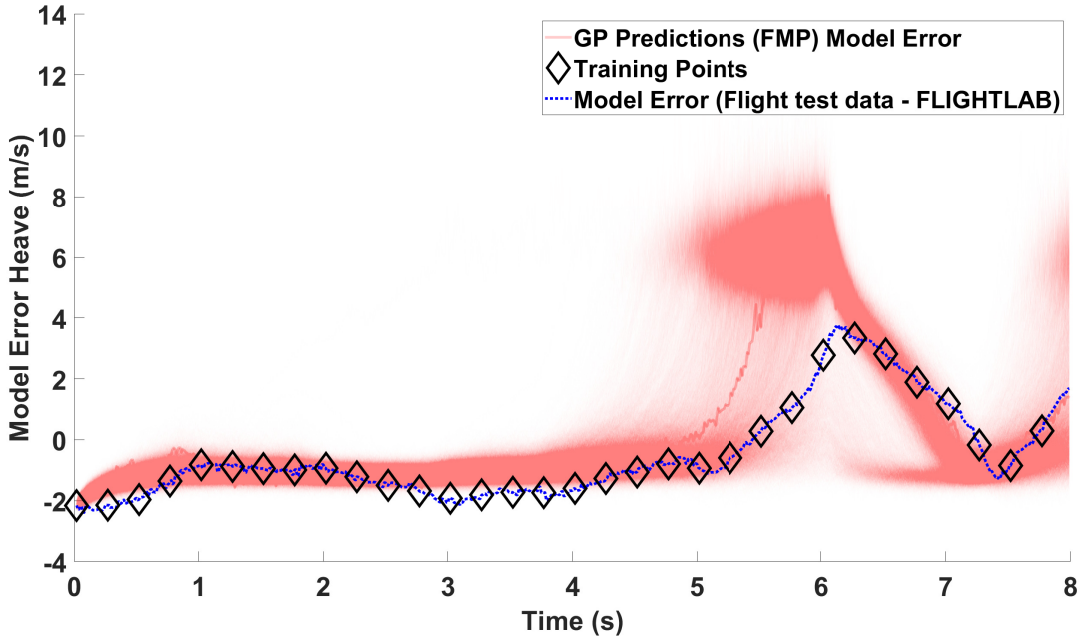


Figure I.85: Realisations of the Gaussian process heave model error full model prediction using the data shown in Figure 9.17 and input structure containing the current collective pilot input with two lagged model error heave terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.72.

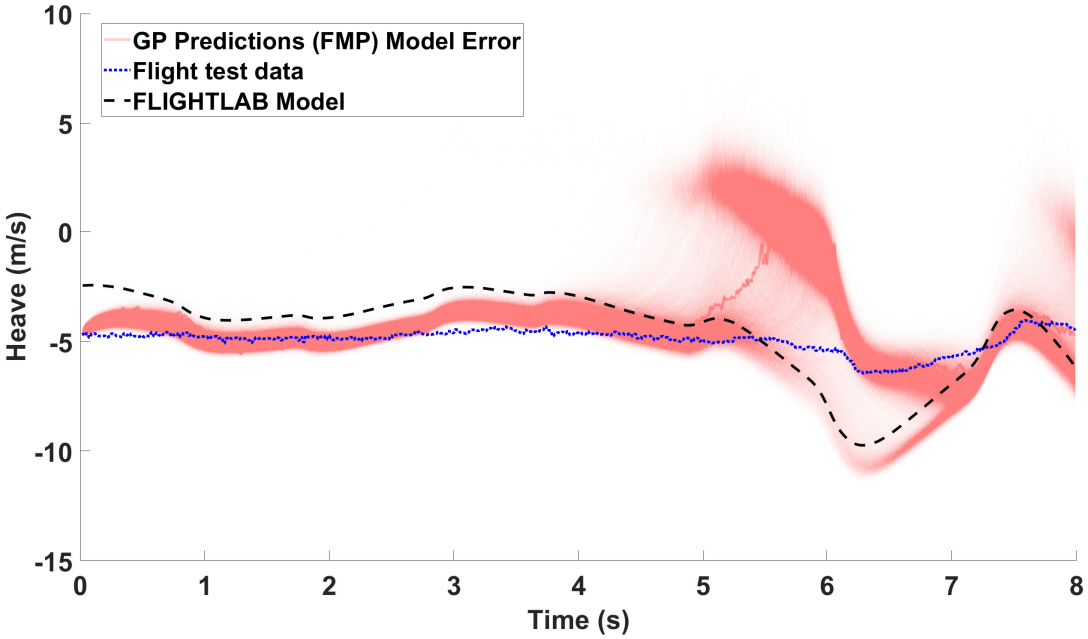


Figure I.86: Realisations of the heave prediction using the model error predictions shown in Figure I.85 with a comparison to the corresponding FLIGHTLAB model.

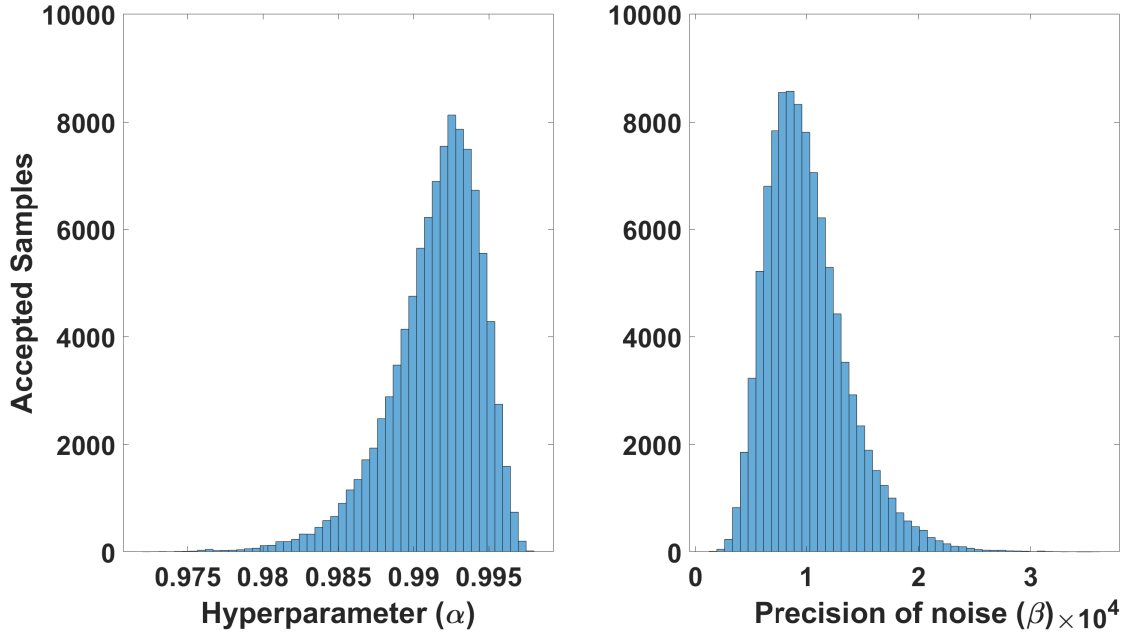


Figure I.87: Simulated annealing results used to create heave model error Gaussian process models, using the input structure containing all of the current pilot inputs with two lagged model error heave terms.

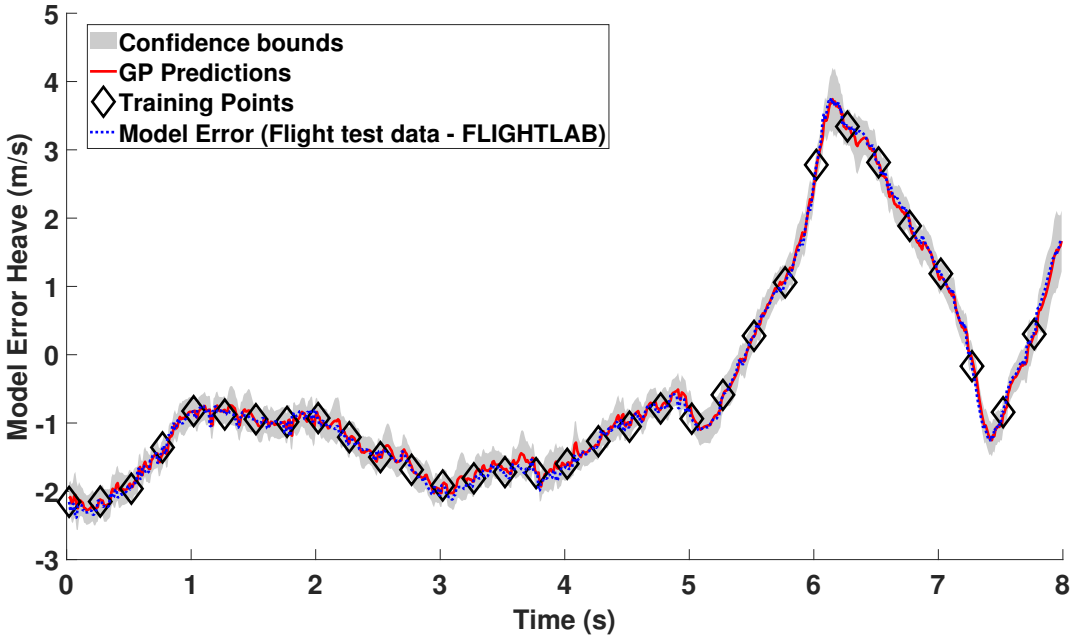


Figure I.88: Gaussian process one step ahead heave model error prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with two lagged model error heave terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.87.

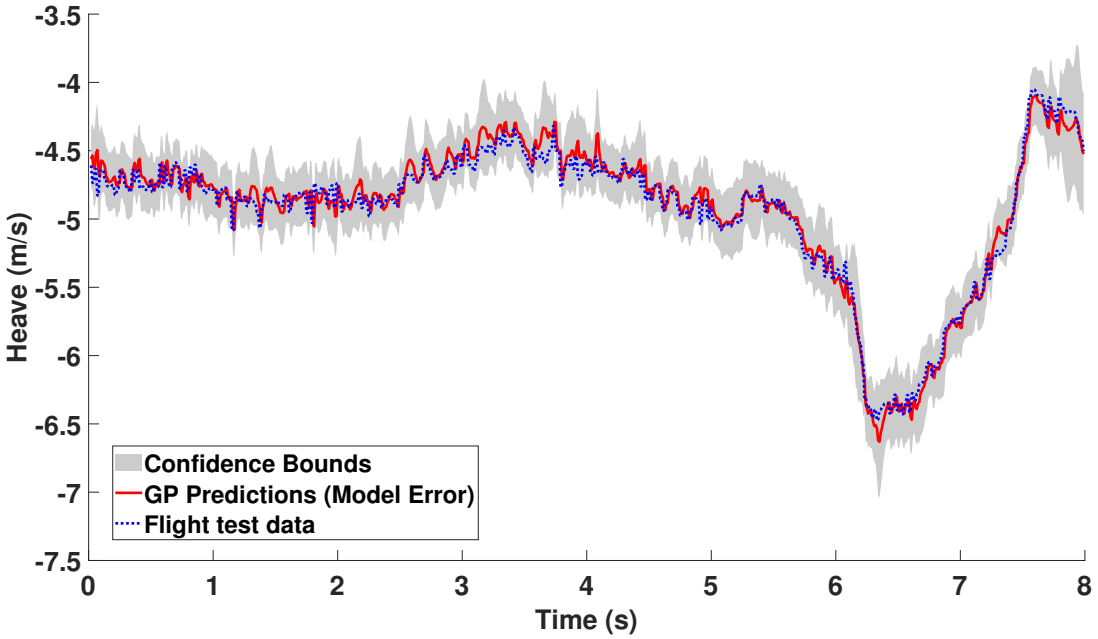


Figure I.89: Gaussian process one step ahead heave prediction using the model error prediction shown in Figure I.88.

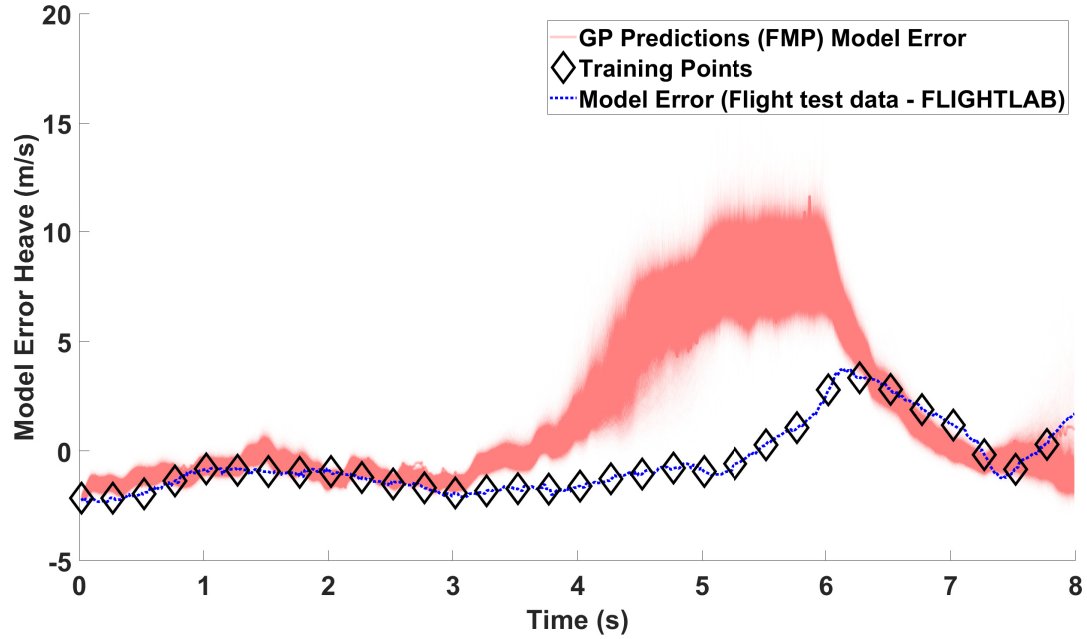


Figure I.90: Realisations of the Gaussian process model error heave full model prediction using the data shown in Figure 9.9 and input structure containing all of the current pilot inputs with two lagged model error heave terms utilising one hyperparameter ($\boldsymbol{\alpha}$) which is located by the simulated annealing results shown in Figure I.87.

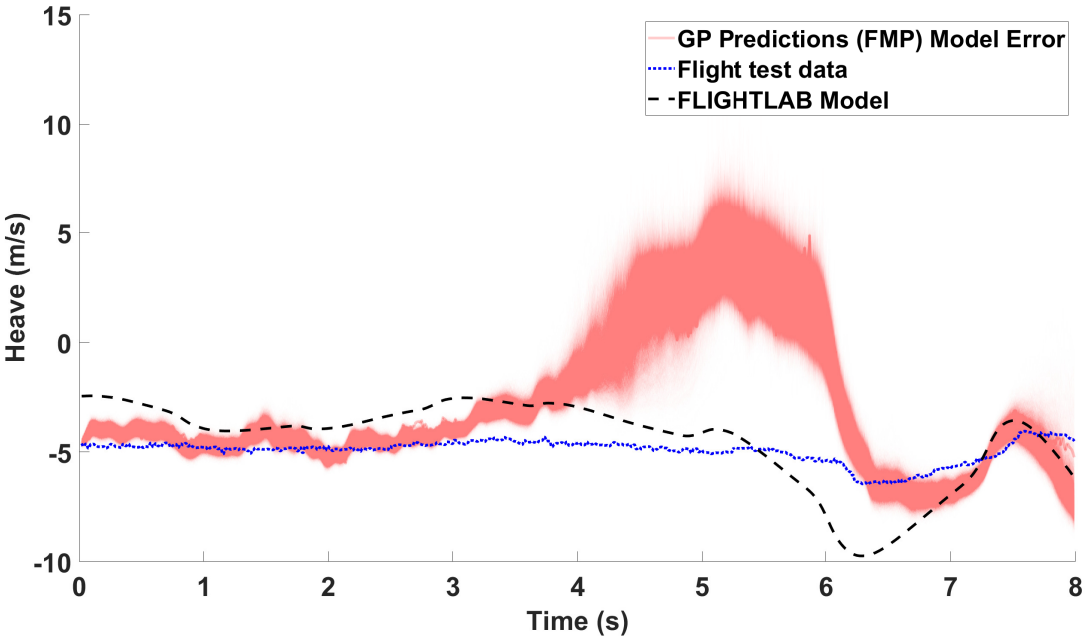


Figure I.91: Realisations of the Gaussian process heave prediction using the model error predictions shown in Figure I.90 with a comparison to the corresponding FLIGHTLAB model.

Bibliography

[1] K. S. Shy, J. J. Hageman, and J. H. Le, "The Role of Aircraft Simulation in Improving Flight Safety Through Control Training," tech. rep., National Aeronautics and Space Administration, Dryden Flight Research Center, 2002.

[2] H. H. Bell and W. L. Waag, "Evaluating the Effectiveness of Flight Simulators for Training Combat Skills: A Review," *The International Journal of Aviation Psychology*, vol. 8, no. 3, pp. 223–242, 1998. DOI: 10.1207/s15327108ijap0803_4.

[3] G. D. Padfield, *Helicopter Flight Dynamics*. John Wiley & Sons, 2008.

[4] A. Punjani and P. Abbeel, "Deep Learning Helicopter Dynamics Models," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3223–3230, 2015.

[5] Anon, "JAR-FSTD H, Helicopter Flight Simulation Training Devices," tech. rep., Joint Aviation Authorities, May, 2008.

[6] Anon, "FAA Advisory Circular AC120-63 Helicopter Simulator Qualification," tech. rep., Federal Aviation Authority, November, 1994.

[7] P. Perfect, E. Timson, M. D. White, G. D. Padfield, R. Erdos, and A. W. Gubbels, "A Rating Scale for the Subjective Assessment of Simulation Fidelity," *The Aeronautical Journal*, vol. 118, no. 1206, pp. 953–974, 2014. DOI: 10.1017/S0001924000009635.

[8] P. Perfect, M. D. White, G. D. Padfield, and A. W. Gubbels, "Rotorcraft Simulation fidelity: New Methods for Quantification and Assessment," *The Aeronautical Journal*, vol. 117, no. 1189, pp. 235–282, 2013. DOI: 10.1017/S0001924000007983.

[9] M. D. White, P. Perfect, G. D. Padfield, A. W. Gubbels, and A. C. Berryman, "Acceptance Testing and Commissioning of a Flight Simulator for Rotorcraft Simulation Fidelity Research," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 227, no. 04, pp. 663–686, 2013. DOI: 10.1177/0954410012439816.

[10] E. Tobias, M. Tischler, T. Berger, and S. G. Hagerott, "Full Flight-Envelope Simulation and Piloted Fidelity Assessment of a Business Jet using a Model Stitching Architecture," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2015.

[11] G. E. Cooper and R. P. Harper, "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities (NASA TN D-5153)," *Moffett Field, CA: National Aeronautics and Space Administration, Ames Research Center*, April 1969.

[12] L. Lu, G. Padfield, M. White, and P. Perfect, "Fidelity Enhancement of a Rotorcraft Simulation Model through System Identification," *The Aeronautical Journal*, vol. 115, no. 1170, pp. 453–470, 2011. DOI: 10.1017/S0001924000006102.

[13] C. K. I. Williams and C. E. Rasmussen, "Gaussian Processes for Regression," in *Proceedings of the 9th Annual Conference on Advances in Neural Information Processing Systems*, 1996.

[14] C. E. Rasmussen, *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*. The University of Toronto, 1999.

[15] C. K. I. Williams, "Regression with Gaussian processes," in *Proceedings of the First International Conference on Mathematics of Neural Networks*, Springer, 1997.

[16] Anon, "FLIGHTLAB Development Software, Advanced Rotorcraft Tehcnology Inc, Sunnydale, California." http://www.flightlab.com/flightlab.html.

[17] M. Titsias, "Variational Learning of Inducing Variables in Sparse Gaussian Processes," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 567–574, 2009.

[18] M. K. Titsias, "Variational Model Selection for Sparse Gaussian Process Regression," tech. rep., The University of Manchester, UK, 2009.

[19] B. Berger, F. Rauscher, and B. Lohmann, "Analysing Gaussian Processes for Stationary Black-Box Combustion Engine Modelling," *IFAC Proceedings Volumes*, vol. 44, no. 01, pp. 10633–10640, 2011. DOI: 10.3182/20110828-6-IT-1002.01160.

[20] S. Castric, L. Denis-Vidal, Z. Cherfi, G. Joly-Blanchard, and N. Boudaoud, "Modeling Pollutant Emissions of Diesel Engine based on Kriging Models: A Comparison between Geostatistic and Gaussian Process Approach," *IFAC Proceedings Volumes*, vol. 45, no. 6, pp. 1708–1715, 2012. DOI: 10.3182/20120523-3-RO-2023.00038.

[21] T. Gutjahr, H. Kleinegräber, H. Ulmer, T. Kruse, and C. Eckstein, "New Approaches for Modeling Dynamic Engine Behaviour with Gaussian Processes," in *Proceedings of the Seventh Conference on Design of Experiment (DoE) in Engine Development*, 2013.

[22] B. Berger and F. Rauscher, "Robust Gaussian Process Modelling for Engine Calibration," *IFAC Proceedings Volumes*, vol. 45, no. 02, pp. 159–164, 2012. DOI: 10.3182/20120215-3-AT-3016.00028.

[23] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 742–747, 2007.

[24] A. O'Hagan, "Curve Fitting and Optimal Design for Prediction," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 40, no. 01, pp. 1–24, 1978. DOI: 10.1111/j.2517-6161.1978.tb01643.x.

[25] R. M. Neal, "Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification," *arXiv preprint*, 1997. arXiv: physics/9701026.

[26] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[27] P. Boyle, *Gaussian Processes for Regression and Optimisation*. PhD thesis, Victoria University of Wellington, 2007.

[28] M. Seeger, "Gaussian Processes for Machine Learning," *International journal of Neural Systems*, vol. 14, no. 02, pp. 69–106, 2004. DOI: 10.1142/S0129065704001899.

[29] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT, 2006.

[30] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, 2003.

[31] M. Ebden, "Gaussian Processes: A Quick Introduction," *arXiv preprint*, 2015. arXiv: abs/1505.02965.

[32] K. Worden, *Nonlinearity in Structural Dynamics: Detection, Identification and Modelling*. CRC Press, 2019.

[33] K. Worden and G. Manson, "On the Identification of Hysteretic Systems. Part I: Fitness Landscapes and Evolutionary Identification," *Mechanical Systems and Signal Processing*, vol. 29, pp. 201–212, 2012. DOI: 10.1016/j.ymssp.2012.01.004.

[34] W. K. Hastings, "Monte Carlo Sampling Methods using Markov Chains and their applications," *Biometrika*, vol. 57, pp. 97–109, April 1970. DOI: 10.1093/biomet/57.1.97.

[35] P. L. Green and K. Worden, "Bayesian and Markov Chain Monte Carlo methods for Identifying Nonlinear Systems in the Presence of Uncertainty," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2051, p. 20140405, 2015. DOI: 10.1098/rsta.2014.0405.

[36] P. L. Green, "Bayesian System Identification of a Nonlinear Dynamical System using a Novel Variant of Simulated Annealing," *Mechanical Systems and Signal Processing*, vol. 52, pp. 133–146, 2015. DOI: 10.1016/j.ymssp.2014.07.010.

[37] P. L. Green, "Bayesian System Identification of Dynamical Systems using Large Sets of Training Data: A MCMC Solution," *Probabilistic Engineering Mechanics*, vol. 42, pp. 54–63, 2015. DOI: 10.1016/j.probengmech.2015.09.010.

[38] J. Ching and Y.-C. Chen, "Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging," *Journal of engineering mechanics*, vol. 133, no. 7, pp. 816–832, 2007. DOI: 10.1061/(ASCE)0733-9399(2007)133:7(816).

[39] J. L. Beck and K. M. Zuev, "Asymptotically Independent Markov Sampling: a New Markov Chain Monte Carlo Scheme for Bayesian Inference," *International Journal for Uncertainty Quantification*, vol. 03, no. 05, pp. 445–474, 2013. DOI: 10.1615/Int.J.UncertaintyQuantification.2012004713.

[40] K. Worden, G. Manson, and E. J. Cross, "On Gaussian Process NARX Models and their Higher-order Frequency Response Functions," in *Solving Computationally Expensive Engineering Problems*, pp. 315–335, Springer, 2014.

[41] K. Worden, C. Surace, and W. Becker, "Uncertainty Bounds on Higher-Order FRFs from Gaussian Process NARX Models," *Procedia Engineering*, vol. 199, pp. 1994–2000, 2017. DOI: 10.1016/j.proeng.2017.09.317.

[42] S. Manso, "Simulation and System Identification of Helicopter Dynamics using Support Vector Regression," *The Aeronautical Journal*, vol. 119, no. 1222, pp. 1541–1560, 2015. DOI: 10.1017/aer.2016.692.

[43] M. V. Kumar, S. N. Omkar, R. Ganguli, P. Sampath, and S. Suresh, "Identification of helicopter dynamics using recurrent neural networks and flight data," *Journal of the American Helicopter Society*, vol. 51, no. 02, pp. 164–174, 2006. DOI: 10.4050/JAHS.51.164.

[44] S. N. Omkar, D. Mudigere, J. Senthilnath, and M. V. Kumar, "Identification of Helicopter Dynamics based on Flight Data using Nature Inspired Techniques," *arXiv preprint*, 2014. arXiv: abs/1411.3251.

[45] A. Girard, C. E. Rasmussen, and R. Murray-Smith, "Gaussian Process Priors with Uncertain Inputs: Multiple-step Ahead Prediction," tech. rep., Univ. Glasgow, Glasgow, Technical Report TR-2002-119, 2002.

[46] A. Girard, *Approximate Methods for Propagation of Uncertainty with Gaussian Process Models*. PhD thesis, The University of Glasgow, 2004.

[47] J. Quinonero-Candela, A. Girard, and C. E. Rasmussen, "Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines-Application to Multiple-Step Ahead Time-Series Forecasting," tech. rep., Technical Report, IMM, Danish Technical University, 2002.

[48] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, "Computer Model Calibration using High-Dimensional Output," *Journal of the American Statistical Association*, vol. 103, no. 482, 2008. DOI: 10.1198/016214507000000888.

[49] B. Dang Vu, "ARISTOTEL Deliverable D2.1 - Baseline Air Vehicle Model Configuration Data." January 2011.

[50] G. D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Oxford, Blackwell Science, 1996.

[51] G. HC(AG06), "Mathematical Modelling for the Prediction of Helicopter Flying Qualities," tech. rep., GARTEUR TP-075, 1996.

[52] R. K. Heffley, J. M. Lehman, and R. A. van Winkle, "A Compilation and Analysis of Helicopter Handling Qualities Data, Volume One: Data Compilation," tech. rep., NASA CR-3144, 1979.

[53] O. Dieterich, J. Götz, M. Pavel, B. Dangvu, P. Masarati, G. Quaranta, M. Gennaretti, J. Serafini, M. Jump, and L. Liu, "Rigid Body and Aeroelastic Rotorcraft - Pilot Coupling - Prediction Tools and Means for Prevention," tech. rep., GARTEUR LIMITED, 2009.

[54] M. Jones, L. Lu, M. Jump, D. Yilmaz, M. Pavel, and B. Dang Vu, "ARISTOTEL Deliverable D2.2 - Rigid-Body Helicopter Model Implementation Report." October 2011.

[55] Anon, "FLIGHTLAB Theory Manual (Vol. One)," tech. rep., Advanced Rotorcraft Technology: Mountain View, California. 132, March 2004.

[56] D. B. Lawrence, *The Flying Qualities of the Wright Flyers*. PhD thesis, Department of Engineering, The University of Liverpool, September 2004.

[57] J. Götz, "Bo105: Flight Test Data," tech. rep., Institute of Flight Systems, Braunschweig, Germany, April 2009.

[58] J. A. Grauer and B. Martos, "Evaluation of Piloted Inputs for Onboard Frequency Response Estimation," in *Proceedings of the AIAA Atmospheric Flight Mechanics (AFM) Conference*, 2013.

[59] P. Hemakumara and S. Sukkarieh, "Learning UAV Stability and Control Derivatives using Gaussian Processes," *IEEE Transactions on Robotics*, vol. 29, no. 04, pp. 813–824, 2013. DOI: 10.1109/TRO.2013.2258732.

[60] S. K. Gan, K. Yang, and S. Sukkarieh, "3d path planning for a rotary wing uav using a gaussian process occupancy map," in *Proceedings of the 2009 Australasian Conference on Robotics and Automation (ACRA)*, 2009.

[61] B. C. Haycock, *Improved Real-Time Helicopter Flight Dynamics Modelling*. PhD thesis, Institute for Aerospace Studies, The University of Toronto, 2015.

[62] M. H. Mansur and M. B. Tischler, "An Empirical Correction Method for Improving Off-Axes Response Prediction in Component Type Flight Mechanics Helicopter Models," tech. rep., NASA-TM-110406, 1997.

[63] J. Quiñonero-Candela and C. E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[64] A. J. Smola and P. L. Bartlett, "Sparse Greedy Gaussian Process Regression," in *Proceedings of the Fourteenth Conference on Advances in neural information processing systems*, pp. 619–625, 2001.

[65] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes using Pseudo-inputs," in *Proceedings of the Nineteenth Conference on Advances in neural information processing systems*, pp. 1257–1264, 2006.

[66] M. Seeger, C. Williams, and N. Lawrence, "Fast Forward Selection to Speed up Sparse Gaussian Process Regression," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS*, 2003.

[67] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proceedings of the 14th Annual Conference on Advances in Neural Information Processing Systems*, pp. 682–688, 2001.

[68] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian Processes for Big Data," *arXiv preprint*, 2013. arXiv: abs/1309.6835.

[69] M. Seeger, "Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations," tech. rep., The University of Edinburgh, 2003.

[70] L. Csató and M. Opper, "Sparse Online Gaussian Processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002. DOI: 10.1162/089976602317250933.

[71] L. Csató, *Gaussian Processes: Iterative Sparse Approximations*. PhD thesis, Aston University Birmingham, UK, 2002.

[72] M. C. Kennedy and A. O'Hagan, "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 03, pp. 425–464, 2001. DOI: 10.1111/1467-9868.00294.

[73] M. Goldstein and J. Rougier, "Probabilistic Formulations for Transferring Inferences from Mathematical Models to Physical Systems," *SIAM Journal on Scientific Computing*, vol. 26, no. 02, pp. 467–487, 2004. DOI: 10.1137/S106482750342670X.

[74] J. Brynjarsdóttir and A. O'Hagan, "Learning about Physical Parameters: The Importance of Model Discrepancy," *Inverse Problems*, vol. 30, no. 11, p. 114007, 2014. DOI: 10.1088/0266-5611/30/11/114007.

[75] C. Zang, H. Xin, and J. Driscoll, "Development and Validation of an Engineering Simulation Model in FLIGHTLAB with Customized Modeling Enhancements," in *Proceedings of the 73rd American Helicopter Society Forum*, May 2017.

[76] H. Xin, C. Zhang, and J. T. Driscoll, "Enhancement of an Engineering Simulation Model to Improve the Correlation with Flight Test Data in Climb/Descent and Autorotation," in *Proceedings of the 75th American Helicopter Society Forum*, May 2019.

[77] S. Conti and A. O'Hagan, "Bayesian Emulation of Complex Multi-output and Dynamic Computer Models," *Journal of Statistical Planning and Inference*, vol. 140, no. 03, pp. 640–651, 2010. DOI: 10.1016/j.jspi.2009.08.006.

[78] T. V. Nguyen, E. V. Bonilla, *et al.*, "Collaborative Multi-output Gaussian Processes," in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, 2014.

[79] M. A. Álvarez and N. D. Lawrence, "Computationally Efficient Convolved Multiple Output Gaussian Processes," *Journal of Machine Learning Research*, vol. 12, no. May, pp. 1459–1500, 2011.

[80] H. Liu, J. Cai, and Y.-S. Ong, "Remarks on Multi-output Gaussian Process Regression," *Knowledge-Based Systems*, vol. 144, pp. 102–121, 2018. DOI: 10.1016/j.knosys.2017.12.034.

[81] G. Parra and F. Tobar, "Spectral Mixture Kernels for Multi-output Gaussian Processes," in *Proceedings of the 30th Annual Conference on Advances in Neural Information Processing Systems*, 2017.

[82] M. Alvarez and N. D. Lawrence, "Sparse Convolved Gaussian Processes for Multi-output Regression," in *Proceedings of the 22nd Annual Conference on Advances in Neural Information Processing Systems*, 2009.

[83] J. Zhao and S. Sun, "Variational Dependent Multi-output Gaussian Process Dynamical Systems," *The Journal of Machine Learning Research*, vol. 17, no. 01, pp. 4134–4169, 2016. DOI: 10.1007/978-3-319-11812-3_30.